Kognitionsvetenskaplig forskning ï *Cognitive Science Research*, ISSN 0281-9864, 2014, 114

Perspektivische Textanalyse
Anweisungen für Vertex

Inger Bierschenk
Bernhard Bierschenk[1]

Abstract Die vorliegende Arbeit ist als Lehr- und Lernbuch gedacht. Der Kern besteht aus Anleitungen zur schrittweisen Einführung in die Anwendung der Perspektivischen Text Analyse (PTA). Diese Schritte sind reichlich illustriert. Um den sprachübergreifenden Charakter von PTA sicherstellen zu können, wurden die Instruktionen an sechs verschiedenen Sprachen geprüft und in der Serie *Cognitive Science Research* publiziert die im Internet unter */archive.org/* zugänglich gemacht wurde. In den gewählten Sprachen (Deutsch, Englisch, Schwedisch, Dänisch, Französisch und Italienisch) handelt es sich um vollständige Aufgabenlösungen. Als wissenschaftlich fundierte Methode für Einzelfall Analysen ist PTA völlig neuartig. Etwas Vergleichbares für das Studium mentaler Prozesse gibt es bislangen nicht. Das geht ganz allein schon aus der Tatsache hervor, dass Doktoranden und Kollegen aus angrenzenden Fachbereichen ihre Beiträge zu wissenschaftlichen Zeitschriften und Dissertationen auf die PTA-Methode aufgebaut haben. Um so viel wie möglich aus dem Übungsgefüge lernen zu können, wird am gewählten Textbeispiel gezeigt werden, wie der vorgegebene Text stufenweise bearbeitet werden kann. Auf dieser Basis lassen sich auch einige Prinzipien für ein prozessorientiertes Vorgehen ablesen. Was mit dem Beispiel aber ganz besonders beabsichtigt wird, ist ein Verständnis für die Bedeutung einer objektiven Textbehandlung aufzubauen. Damit ist gemeint, dass die einzelnen Schritte so ausgeführt werden müssen, dass eigene Vorstellungen über den Inhalt des vorgegebenen Textes die Arbeit nicht beeinträchtigen können.

Abstract The present work is conceptualised as textbook. Its core consists of instructions to a stepwise introduction of Perspective Text Analysis. The steps are richly illustrated. To secure the inter-lingual character of PTA, the instructions have been tested in the context of six different languages. Furthermore, the complete solutions to the tasks have been proven with respect to the selected languages (German, English, Swedish, Danish, French, and Italian). These versions have previously been published with */archive.org/*. As scientific method in the true sense of the notion science, PTA represents a completely new approach. Thus far, to our knowledge, something comparable for the study of mental processes does not exist. This circumstance is easily verified through the fact that students and colleagues from neighbouring disciplines have based their contributions to Journals as well as their doctoral theses on the PTA-method. To profit as much as possible from the configuration of the exercises, the approach to the text example is designed for step-by-step processing. Against this background, it is possible to bring out a number of principles for semantic-free and process-oriented procedures. Of particular significance is our aim to build up an understanding of the import of objective text processing. This means that the single steps should be carried out in a way that circumvents those imaginations and prejudices about the content of a text enter into the analysis.

Text als Organisation und Struktur

Textannäherung

Was ein Text aus einem bestimmten Forschungsbereich ausdrückt ist im Wesentlichen das Resultat der Vorstellungen einzelner Wissenschaftler. Aus einer breiteren sozialwissenschaftlichen Perspektive gesehen, existieren deswegen genau so viele Methoden zur Textanalyse wie es Projekte oder Forschungsfelder gibt, mit zum Beispiel sozialen, psychologischen, ökonomischen und politischen Begrenzungen. Dieser Umstand ist ein

[1] Contact: Bernhard Bierschenk, Department of Psychology, Lund University, Box 213, SE-221 00 Lund, Sweden; information: https://archive.org/details/studiesinconsciousness

Resultat der Art und Weise wie einzelne Forscher oder Forschungsgruppen ihre eigenen Theorien über die *Struktur* eines Feldes entwickeln. Die Strategie der Inhaltsanalytiker geht zunächst darauf hinaus, mit der angewandten Methode einen bestimmten Ansatz zu bekräftigen oder, in einzelnen Fällen eine Struktur zu verwerfen. Die Methode der *Inhaltsanalyse* und ihre Anwendung wurden unter anderem in Bos und Tarnai (1989, 1996) diskutiert.

Inhaltsanalysen jeglicher Art enthalten a priori definierte Schemata für Klassifikationen, die vom Inhaltsanalytiker mit mehr oder weniger zuverlässigen Resultaten auf Texte appliziert werden (B. Bierschenk, 1978; B. Bierschenk, & I. Bierschenk, 1976). Der ganze Applikationsprozess ist sehr zeitaufwendig und wird mit wachsenden Textmaterialien nahezu unübersteiglich. Wenn nun aber einmal eine Analyse zustande gekommen ist, wird es dennoch kaum möglich sein, diese Leistung noch einmal zu wiederholen, unter anderem deswegen, weil der Inhaltsanalytiker kaum in der Lage sein wird, seine, auf frühere Arbeitsgänge angewandten Klassifikationen, im Gedächtnis behalten zu können. Schließlich ist es außerdem sehr schwierig, die gleichen Beurteilungskriterien beizubehalten, und darüber hinaus auch noch eine perfekte Übereinstimmung mit anderen Beurteilern zu erreichen. Deswegen sind die Methoden der Inhaltsanalyse mit einer Reihe von statistischen Zuverlässigkeitstests versehen, was aber notwendigerweise dazu führen muss, dass inhaltsanalytische Resultate immer von Wahrscheinlichkeitsschätzungen abhängig gemacht werden müssen.

Eigene Erfahrungen mit großen Mengen von Interviewdaten (4000 Seiten!), die freie Interviewfragen beinhalten, wurden seit den 1970iger Jahren inhaltsanalytisch bearbeitet (I. Bierschenk, 1977). Welche Mühen mit einer jahrelangen Entwicklung von Regelsystemen, zusammenhängen, wurde in B. Bierschenk (1978) nachgewiesen. Auf den Erfahrungen mit dem Verfassen von Anweisungen und der Schätzung von Übereinstimmungen zwischen unterschiedlichen Beurteilern sei es erlaubt, festzustellen, dass die dabei herauskommenden Resultate nichts anderes ergeben, als das Gefühl eine Menge über Textproduktionen und Design gelernt zu haben. Einerseits haben diese Arbeitsprozesse dazu beigetragen, eine tiefere Einsicht in die einzelnen Analysenschritte zu schaffen, andererseits wurde es aber immer schwieriger beobachten zu können, wie eine Textgestaltung oder textuelle Erscheinung in der Lage sein könnte, etwas Internes, d. h., Struktur widerspiegeln zu können.

Es ist allgemein bekannt, dass sich Sprach- und Sozialwissenschaftler, aus klassischer Sicht gesehen, in ihren Textbehandlungsstrategien grundlegend unterscheiden. Für einen Linguisten inkludiert zum Beispiel ein Text immer auch seinen Kontext. Für einen Sozialwissenschaftler bedeutet Kontext dagegen ein breiterer Interpretationsrahmen was mit sich bringt, dass Text in einen Kontext einbezogen ist. Irgendwo zwischen diesen Grenzen treffen sich beide unter einem gemeinsamen Banner, nämlich unter dem Strukturbegriff. Schließlich verstehen beiden diesen Begriff als Ausdruck einer inneren Textqualität, die sie einzufangen versuchen indem sie sich mit der Textoberfläche, d. h. mit der Organisation, befassen. Inhaltsanalyse und Semantik beziehen sich auf die gleiche Grundidee, was sich als Konsequenz aus dieser Auffassung ergibt.

Text als Organisation

Sowohl die Definition von Einheiten und deren Relationen als auch ihre Beschreibung in Stufen gibt die Grundlage für Organisationen ab. Die Identifikation von Einheiten geht gewöhnlich von einer Basiseinheit aus während die Analyse von Stufen durch die Anwendung von Regelsystemen gesteuert wird. Dabei werden die Einheiten assoziativ in eine serielle Ordnung gebracht. Damit können effektive Verfahren zur Erstellung von mehr oder weniger vollständigen Überblicken wirksam werden. Der Weg in einer geschaffenen

Baumstruktur kann sich dabei von der Basis bis hin zum Gipfelpunkt mehr oder weniger geradlinig gestalten.

Abhängig von der organisatorischen Komplexität der begründeten Relationen kann sich der Pfad zum Gipfel in eine Reihe von parallelen Serien verzweigen und sich auf Zwischenstufen verwirklichen. Ein und dieselbe Einheit kann die gleiche Funktion haben, aber auf verschiedenen Stufen unterschiedlich beschrieben werden. Für das Endresultat stellen parallel verlaufenden Serien unterschiedliche Gewichte dar. Manchmal müssen aber auch Entscheidungen getroffen werden, die mit der Funktion einzelner Einheiten zusammenhängen. Einheiten können damit als Schleusen im Aufbau einer Funktion auftreten oder auch nur als Füllungen oder Barrieren auftreten. Gemeinsam für Organisationen unterschiedlicher Art ist es, dass diese sowohl physisch als auch materiell zugänglich sein müssen, was bedeutet, dass sie konkrete Stellvertreter haben müssen, die verdeckte Teile leicht erreichbar machen, z. B., durch Referenz und Inferenz.

Eine Konsequenz der sozialwissenschaftlichen *Standardmethode* (Pinker, 2002) ist es, dass Text als Organisation und als kontextueller Teil einer Ansammlung von Organisationen betrachtet werden muss. Ein solche Betrachtungsweise lässt sich mit Agentenschaft (= Agency) umschreiben. Diese Benennung bezieht sich auf die Tatsache, dass die *Agenten* in einem Text keine autonome Ganzheit bilden. Die Bedingung Agency kommt vielmehr durch Verkettungen zustande. Auch wenn die Verkettungen nur willkürliche Ordnungen darstellen, ist die Agentenschaft eines Texts der gemeinsame Nenner, den eine Analyse zu entdecken hat.

Aus einem engeren linguistischen Kontext gesehen ist Organisation gleichbedeutend mit einem grammatischen Satzgefüge und dessen Teile. Die Hauptkomponenten sind die bestimmenden Teile, die anderen sind die unterordneten Komponenten. Die Mehrheit der Sozialwissenschaftler ist der Überzeugung, dass sie mit dem Begriff Struktur die Ordnung und Ordnungsrelationen ansprechen, genauso wie es die Sprachwissenschaftler getan haben, indem sie ein Satzgefüge mit *syntaktischer Struktur* beschreiben. Nun stellt sich aber die Frage, wie die Vermischung von Organisation mit Struktur erklärt werden kann. Klassische Linguistik nimmt an, dass ein Kernsatz den Ausgangspunkt für Expansionen und Transformationen bilden kann, die außerdem beobachtbar sind. Transformation bezieht sich auf einen strukturellen Begriff, wird aber in der Satzanalyse angewandt, um positionelle Veränderungen zu beschreiben, z. B., die Passivkonstruktion.

Während der 1960iger und 70iger Jahre verursachte die Vorstellung, dass eine strukturelle Entwicklung durch syntaktische Transformationen nachgewiesen werden könnte, eine Flut von psycholinguistischen und psychologischen Tests. Eine der Hypothesen, die getestet wurde, galt der Transformation eines Kernsatzes. Es wurde untersucht ob und inwieweit der Grad der Transformation sich im Schwierigkeitsgrad des Verstehens widerspiegeln könnte. Verständnis wurde anhand von Bearbeitungszeiten im Lesen von Sätzen gemessen. Die Bearbeitungszeiten konnten keinen erkennbaren Effekt zum Ausdruck kommen lassen. Zusammenfassend kann also festgestellt werden, dass organisatorische Veränderungen keine strukturelle Kontrolle garantieren können, unbeschadet dessen, ob die Steuerungskomponente aus Text besteht oder nicht.

Text als Struktur

Erfahrungen mit Organisationen weisen oftmals darauf hin, dass informelle Wege viel wichtiger seien können als die Formalstufen auf der Treppe zur Spitze. In Verhältnissen, in die Menschen einbezogen sind, gibt es Operationen, deren Dynamik sowohl regelbrechend als auch unerwartet sein kann. Verbunden mit dynamischen Funktionen sind vor allem offene Kanäle. Um in einem System Wachstumsentwicklungen garantieren zu können, müssen Möglichkeiten für positionelle Veränderungen vorliegen. Im Hinblick auf die Sensibilität geistiger Kräfte müssen Strömungen kanalisiert werden können.

Im Ganzen gesehen scheint es so als ob das beste Resultat erzielt werden kann, wenn wenigstens einige Positionen mit Schlüsselfunktionen gefüllt werden. Wenn es um Klarheit über die Wege einer Organisation und deren Ziele geht sollten sich diese Funktionen nicht ändern. Unter dieser Voraussetzung stützen Position und Funktion einander. Folgerichtig ist eine Struktur immer nur teilweise sichtbar und durch organisatorische Schlüssel oftmals nur zufällig erkennbar. Da sie aber stabil ist, lässt sich der Puls, die Bewegung oder die Stärke wahrnehmen, ohne die eine Organisation nichts anderes wäre als eine leere Schale.

Eine *Textanalyse*, deren steuernder Begriff Struktur ist, muss in der Lage sein, zwischen Funktionen und Kanälen Ordnung halten und der steuernden Komponente (der Quelle) Operationsgewicht geben können. Unterschiedliche sozialwissenschaftliche Modelle benutzen den Begriff *Akteure*, der manchmal für Personifizierung (im Sinne von einem einzelnen Repräsentanten) oder Organisation angewandt wird. Typisch für einen Akteur ist einzig und allein Repräsentation. Funktion und Position sind damit getrennt nicht zugänglich. Folglich ist für ein solches Modell der Begriff *Rolle* zentral. Die gleiche Idee liegt dem linguistischen Modell zugrunde. Eine sogenannte grammatische Rolle baut auf ein positionelles Denken auf, wo Position mit Rolle gleichgestellt wird und wo verschiedene Repräsentanten miteinander gleichgestellt sind. Gleichheit wird durch Klassifikationen garantiert, d. h., die Repräsentanten müssen vom gleichen Typ sein, um ins Rollenmodell zu passen. Eine Analyse in diesem Sinne gründet sich auf *szenisches* Vorhandwissen, das den Rahmen für die Rollen abgibt (I. Bierschenk, 1984).

In einem Strukturmodell muss dagegen eine *Perspektive* entdeckt werden können. Nur dann können Quelle und Agent in einem angularen Vernetzungsprozess auseinandergehalten werden. Dies setzt nicht nur Repräsentation, sondern vor allem auch Individualität voraus. Um klarstellen zu können, auf welche Weise eine Perspektive in einem Text vorhanden ist, muss der Begriff Akteur mit dem Begriff Agent ersetzt werden. Agent steht nämlich nicht nur für eine individuelle sondern auch für ene autonome Prozesssteuerung. Worum es in einem Prozess geht, lässt sich da ablesen wo die Steuerung sichtbar wird. Ihre Quelle manifestiert sich in dem meist geeigneten Gebilde: *Form*. Deswegen kann die Gestalt einer Subkomponente oder ein Teil, der die Ganzheit verkörpert, während der Verarbeitung in allen Teilen unbekannt bleiben.

Dass die Agentenfunktion an die erste Position im *funktionellen Satz* gebunden ist, ist eine wichtige Konsequenz (I. Bierschenk, 1999, 2011a). *Vertex* verwendet funktionell gebundene Positionen und entdeckt, inwieweit diese gefüllt sind. Wenn sie leer sind, sind die Variablen nicht sichtbar. Sind sie dagegen gefüllt, beinhalten sie Variablen oder sichtbare repräsentative Agenten. Leere Positionen bedeuten also, dass sich der Agent versteckt hält, aber gleichzeitig ein *Schlüsselloch* öffnet, wodurch es möglich wird, in den Textraum schauen zu können. In Anlehnung an das oben angeführte Kanalisierungsbeispiel lässt sich feststellen, dass diese Schlüssellöcher den Fluss von Informationen durch den Text sichtbar machen. Wenn das der Fall ist, steuert der unbekannte Agent von einem anderen Platz. Er kann aber durch das Loch entdeckt werden.

Eine gebundene Position wird zu einem *Fenster*, durch das der Textproduzent sich zeigen kann indem eine *Schattengestalt* sichtbar wird, sich voll entwickeln kann und am Ende fassbar wird. Die veränderlichen Agenten, die sogenannten *textuellen Agenten*, haben eine Funktion gemeinsam, nämlich den Informationsfluss zu kanalisieren, der vom unbekannten Agenten ausgeht. Irgendeine andere Ähnlichkeit lässt sich a priori unmöglich angeben. Somit ist es die funktionelle Anwendung des Agenten, die den Weg zu perspektivischen Informationen ebnet. Die Relation zwischen textuellen Agenten, Textproduzent und irgendeiner übergeordneten Idee ist auf der einen Seite unabhängig von Klassen, auf der anderen aber abhängig von Kategorien im kantischen Sinn.

Textbearbeitung

Seit den 1980iger Jahren wurde die Methode für eine perspektivische Textanalyse sowohl in Kursen und Konferenzen als auch in Sammelwerken dargestellt (z. B. in B. Bierschenk, 1984, 1991, 1993; B. Bierschenk & I. Bierschenk, 1993; B. Bierschenk, I. Bierschenk, & H. Helmersson, 1996). Weiterhin liegt der Entwicklung Anwendererfahrungen zugrunde (I. Bierschenk, 2011b). Insbesondere soll noch einmal darauf hingewiesen werden, dass Vertex eine Weiterentwicklung von PTA ist und damit strikte Messungen von angularen Veränderungen einbezieht. Diese haben dazu beigetragen, nichtlineare Beschreibungen von sprachlichen Energieflüssen geben zu können. Das Ergebnis wurde in Form von Energielandschaften dargestellt. (I. Bierschenk, 2011 a; I. Bierschenk & B. Bierschenk, 2004, B. Bierschenk, 2011).

Die hier vorbereiteten Anweisungen zur Anwendung der Methode sollen nun Schritt für Schritt aufzeigen, wie sich Energielandschaften bilden und Gestalt annehmen. Da das Interview eine in den Sozialwissenschaften häufig angewandte Methode für die Datenerhebung ist, soll ein Textabschnitt aus einem Interview die Grundlage dafür abgeben wie eine Perspektive dargestellt werden kann. Die Benennung *Perspektive* kann natürlich gegen *Intention*, *Motiv*, *Interesse*, *Persönlichkeit* oder etwas Anderem mit immaterieller Bedeutung ausgetauscht werden.

Die Anweisungen sind so aufgebaut, dass der Benützer einem Verfahren folgt, das in einer Anzahl von Schritten deutlich macht, was mit einem Text geschieht, der auf geometrischer Grundlage bearbeitet wird. Das Verfahren besteht aus fünfzehn Schritten, die in folgenden Themen gegliedert sind: (1) Textbehandlung, (2) angulare Berechnung und (3) geometrische Repräsentation. Schließlich folgt auch noch ein Exkurs, der die (4) Benennungsprozesse veranschaulicht.

Die dargestellten Übungen sollen ein Verständnis für die Bedeutung geometrischer Überlegungen schaffen. Damit ist gemeint, dass die Anwendung von PTA/Vertex sicherstellen soll, dass eigene Haltungen sowie Motive und Perspektiven aus dem Verfahren ausgegrenzt werden. Auch wenn es hier um Textanalysen geht, ist es wichtig nochmal darauf hinzuweisen, dass Vertex frei von semantischen Überlegungen ist und ausserdem keine besonderen grammatischen Kenntnisse voraussetzt. Alle notwendigen Kenntnisse lassen sich leicht in einem gewöhnlichen Lexikon finden. Zur Demonstration wird folgender Wortlaut angeboten:

> *Schauen Sie sich die Einstellung von heutzutage an, und es ist ja nicht nur unter den Gemeindeangestellten, die meisten meinen eben, ich habe ja mein Gehalt, wieso soll ich denn mithelfen, Wege zu finden(,) wie die Gemeinde Geld sparen könnte, das ist mir doch völlig egal. Es ist das gleiche Denken hier.*

Im gegebenen Fall handelt es sich um einen Gemeindeangestellten, der für ökonomische Transaktionen verantwortlich ist und seine Arbeitssituation in einem Interview darstellt. Die Formulierung stammt aus einem Forschungsmaterial, das von Studenten der Ökonomie und Betriebswirtschaft erhoben worden ist und in einem Doktorandenseminar an der Universität in Lund, Schweden, bereitgestellt wurde. Der Text ist also gerade so sprachrichtig, wie man es typischerweise von Interviews erwarten kann. Die Bedeutung des Texts bezieht sich auf den Stil. Der Stil spiegelt den Grad des Unausgesprochenen oder der Texttiefe wider. Eine erste Maßnahme sollte daher sein, sich mit dem dargebotenen Übungstext vertraut zu machen.

Übung 1: Transposition

Die folgende Aufgabe besteht darin, den Text so zu verändern, dass sich das Textformat für ein Computer-basiertes Verfahren eignet. Um die dafür notwendigen Daten erhalten zu können, muss der Text erst einmal auf seine Hochkante gestellt werden. Damit erhält die Form, die in Tabelle 1 dargestellt wird.

Tabelle 1

Transposition

Zeile	*Text*	*Zeile*	*Text*	*Zeile*	*Text*
1	[.]	23	meisten	44	Gemeinde
2	*	24	meinen	45	Geld
3	Schauen	25	Eben	46	sparen
4	Sie	26	,	47	könnte
5	sich	27	ich	48	,
6	die	28	habe	49	das
7	Einstellung	29	ja	50	ist
8	von	30	mein	51	mir
9	heutzutage	31	Gehalt	52	doch
10	an	32	,	53	völlig
11	,	33	wieso	54	egal
12	und	34	soll	55	.
13	es	35	ich	56	*
14	ist	36	denn	57	Es
15	ja	37	mithelfen	58	ist
16	nicht	38	,	59	das
17	nur	39	Wege	60	gleiche
18	unter	40	zu	61	Denken
19	den	41	finden	62	hier
20	Gemeindeangestellten	42	wie	63	.
21	,	43	die	64	[*]
22	die				

Wie aus Tabelle 1 hervorgeht, ist jede Zeile mit einer Buchstabensequenz von unterschiedlicher Länge besetzt. Einige bestehen aus einem einzigen Buchstaben (= Graphem) wie zum Beispiel (,) und (.), andere haben mehrere. Im weiteren Verlauf sind es die Grapheme und die Grenzmarkierungen, die den Prozess steuern werden. Jedes Wort, d. h., jede Sequenz erhält also seine eigene Zeile, was auch für Trennungszeichen zweiter Ordnung (,) gilt. Falls sich zu Beginn und am Ende keine natürliches Trennungszeichen erster Ordnung (.) finden lässt, muss der Text außerdem noch mit einer technischen Markierung [.] versehen werden, die Beginn und Ende signalisiert. Innerhalb dieser äußeren Grenze sind überdies auch Startmarkierungen (*) notwendig, die den Beginn einzelner Sätze anzeigen.

In Tabelle 1 erweitert sich der Text also mit zwei (2) Zeilen zu Beginn und einer (1) Zeile am Schluss sowie einer (1) Zeile vor dem letzten Satz. Ein geeignetes Programm, das diesen Schritt erleichtern kann, ist z. B. Excel. Seine Flexibilität, im Hinblick auf den Aufbau von Tabellen, macht es möglich, einen Text im Codierungsprozess nach und nach mit Leerzeilen und -kolonnen zu erweitern. Die Zeilen gleich am Anfang zu nummerieren, ist weiterhin eine sinnvolle Massnahme. Nach diesem ersten Schritt sollte nun ein Resultat vorliegen, dass mit der Aufstellung in Tabelle 1 übereinstimmt.

Übung2: Lexikalische Codierung

In der *Lexikonanwendung* wird es notwendig, die Zusammensetzung solcher Graphemstrenge festzustellen, die aus dem Lexikon der deutschen Sprache hervorgehen. Nachdem jeder Strenge lexikalisch bestimmt worden ist, wird es notwendig, den einzelnen Strengen Codes zuzuordnen. Diese werden in Tabelle 2 vorgestellt. Wie aus der Tabelle ersichtlich ist, bestehen die Codes aus einem System mit zwei Ziffern und zwar von 00 bis 90, was bereits in Bierschenk und Bierschenk (1976) vorgestellt und diskutiert wurde. In Tabelle 2 wird die Zugehörigkeit erklärt während die für den Beispieltext bestimmenden Markierungen in Tabelle 3 zugänglich gemacht werden.

Tabelle 2

Codesystem

Feststellung	*Symbol*	*Beschreibung*	*Code*
Technischer Satzmarkör	[1][.]	Technische Einfügung einer Periode	00
Satzmarkör 2. Grades	dass	Natürlich auftretende Grenzmarköre	01
Agentenvariable	A_X	Kontext- oder konditionale Restriktion	10
Agentenvariable	A_X	Erfahrungsbedingte Spezifizierung	20
Manifestierter Agent	A	Explizit ausgedrückter Agent	30
Impliziter Agent	*	Implizit und unbedingter Agent	30
Verb	ω	Nukleus eines Kernsatzes	40
Objektiv	O	p_0 = ohne Zeiger und explizit gegeben	50
Objektiv	*	p_0 = ohne Zeiger und implizit gegeben	50
auf-Objektiv	-	p_1 = Zeiger auf	60
mit-Objektiv	-	p_2 = Zeiger mit	70
für-Objektiv	-	p_3 = Zeiger für	80
Phrase	-	Verb-lose Strenge, die einem Markör folgen	90
Technischer Satzmarkör	*	Einfügung vor dem Ende einer Periode	01
Satzmarkör 1. Grades	.	Natürlich auftretende Satzmarköre 1. Grades	00

[1]Satzmarkör ersten Grades (Sentence Marker (SM)); [2]Satzmarkör zweiten Grades (Clause Marker (CM))
Kommentar: Das Einsetzen von technischen Satzmarkören [**.**] geschieht immer dann, wenn funktionelle Sätze implizit sind, was weiter unten illustriert wird.

Tabelle 3

Empirisches Lexikon

Verb (40)	*Prä (60)*	*Prä (70)*	*Prä (80)*	*[2]Satz (01)*	*[1]Satz (00)*
Schauen	von	-	-	die	.
ist	an			,	
meinen	unter			und	
habe	zu			wieso	
soll				denn	
mithelfen				wie	
finden					
sparen					
könnte					

[1]Satz ersten Grades (Sentence Marker (SM)); [2]Satz zweiten Grades (Clause Marker (CM))

Kommentar: Als Verb werden alle Formen in Rechnung gezogen. Gültige Formen bauen auf erkennbare Grundformen auf, was Konsequenzen für ihre Bedeutung im Codierungsprozess haben wird.

Die entwickelten Routinen müssen die Zugehörigkeit folgender Typen markieren können: Satzmarkierungen ersten Grades (SM) [**.** ! ?], und Satzmarkierungen zweiten Grades (CM) (, : ; -) als auch die bereits genannten technischen Satzmarkierungen wie [**.**] und [*] sowie natürlich vorkommenden Satzmarkierungen (wie z. B. *und*, *dass, weil*) und Präpositionstypen (*auf*, *mit* und *für*). Schließlich muss das Lexikon alle für den zu bearbeitenden Text relevanten Verben enthalten. Der erste Typ der Grenzmarkierungen hat also insgesamt drei Fälle, während der zweite Typ mehrere Trennungszeichen für die Periodisierung erkennt.

Mit dem Ausgangspunkt in der Lexikoncodierung beginnt der Prozess von hinten oder unten und sucht seinen Weg nach vorne oder oben. Diese Prozessordnung wurde bereits in den 80iger Jahren empirisch begründet (Bierschenk & Bierschenk, 1986 a, b). Nun sollen alle Strenge, die bisher keine Codes bekommen haben, erkannt werden und ihre Codes zugeteilt bekommen.

Die Zuteilung bedeutet eine lexikalische Bestimmung der einzelnen Strenge. Die Lösung ergibt sich aus Tabelle 4, die den gesamten Beispielstext auf 64 Zeilen abträgt, dabei aber nur 34 markierte Zeilen beinhaltet. Wie aus Tabelle 4 ersichtlich ist, werden die Codes der gefunden Strenge am besten links vor dem jeweiligen Textstreng markiert. Die Textlänge hat sich außerdem etwas erweitert.

Tabelle 4
Lexikale Codierung

Zeile	*Code*	*Streng*	*Zeile*	*Code*	*Streng*	*Zeile*	*Code*	*Streng*
1	00	[.]	23	-	meisten	45	-	Geld
2	01	*	24	40	meinen	46	40	sparen
3	40	Schauen	25	-	eben	47	40	könnte
4	-	Sie	26	01	,	48	01	,
5	-	sich	27	-	ich	49	-	das
6	01	die	28	40	habe	50	40	ist
7	-	Einstellung	29	-	ja	51	-	mir
8	60	von	30	-	mein	52	-	doch
9	-	heutzutage	31	-	Gehalt	53	-	völlig
10	60	an	32	01	,	54	-	egal
11	01	,	33	01	wieso	55	00	.
12	01	und	34	40	soll	56	01	*
13	-	es	35	-	ich	57	-	Es
14	40	ist	36	01	denn	58	40	ist
15	-	ja	37	40	mithelfen	59	-	das
16	-	nicht	38	01	,	60	-	gleiche
17	-	nur	39	-	Wege	61	-	Denken
18	60	unter	40	60	zu	62	-	hier
19	-	den	41	40	finden	63	00	.
20	-	Gemeinde angestellten	42	01	wie	64	01	[*]
21	01	,	43	01	die			
22	01	die	44	-	Gemeinde			

Übung 3: Implizite funktionelle Sätze

Damit die fortgesetzte Codierung korrekt ausfällt, müssen alle *funktionellen Sätze*, auch solche, die bisher nicht markiert und damit unerkannt und implizit sind, festgestellt werden können. Der funktionelle Satz besteht aus folgendem Kern:

Agent + Verb + Objektiv

In Übereinstimmung mit dieser Formel kann ein *funktioneller Satz* nur ein Verb haben, und zwar nur ein einziges! Die Sätze sollten im Prinzip so markiert werden, wie aus Tabelle 5 hervorgeht.

Tabelle 5
Das Prinzip für funktionelle Sätze

Code	*Komponente*
01	*
30	*
40	**Verb**
50	*
01	*
30	*
40	**Verb**
50	*
01	*

Für die Erkennung eines funktionellen Satzes muss sich zumindest ein Streng sowohl vor als auch hinter dem Verb realisiert haben. Können keine Strenge vor beziehungsweise hinter dem Verb ausgemacht werden, muss ein Platzhalter (*) für die fehlenden Agenten- und Objektivstrenge eingeführt werden. Da aber nur ein Verb vorkommen darf, müssen auch weitere Grenzmarköre eingesetzt werden, die die Aufgabe haben, die Reichweite für jedes einzelne Verb abzugrenzen. Dieses Prinzip kommt immer dann zur Anwendung, wenn die Routinen nach oder vor dem Verb auf eine Satzmarkierung treffen, die mit sich bringt, dass eine Leerzeile für den jeweiligen Platzhalter (*) eingesprengt werden muss.

Kommentar: Mit diesem Prinzip ist die Grundlage gegeben, durch eine algorithmische Bearbeitung mehrere Satzmarkierungen einsprengen zu können. Zu einem späteren Zeitpunkt werden die durch (*) markierten Stellen mit spezifischen Informationen ausgetauscht, was aus der Aufgabe in Übung 5 hervor gehen wird. Die Formel für *Verb-lose Strenge* lautet:

> *Strenge, die kein Verb einschließen, sich aber zwischen zwei Satzmarkierungen befinden, stellen das Supplement für das Objektiv (*) oder den Agenten dar.*

Übung 4: Algorithmische Codierung

Nach der Codierung mit dem Lexikon folgt also ein Schritt, der es erforderlich macht, die Anzahl der Sätze so zu markieren, dass jeder Satz aus einer (A)- und einer (O)-Komponente besteht. Um diesen Umstand festhalten zu können, müssen die (A) und (O) Einheiten mit einer Blockbezeichnung (B) markiert werden. Diese Codierung macht eine eigene Kolonne erforderlich, was in Tabelle 6 deutlich wird. Die Blocknummern in laufender Ordnung beginnen an der Stelle, wo der natürliche Satzbeginn liegt.

Der eigentliche algorithmische Prozess muss immer und ausnahmslos mit dem Textschluss beginnen. Nun können alle übrigen, d. h. die lexikalisch *nicht-definierten* Strenge, erkennbar gemacht werden. Wie bereits angeführt wurde, ist die algorithmische Bearbeitung eines Texts empirisch begründet worden (Bierschenk & Bierschenk, 1986 a, b) was nun demonstriert und kommentiert werden soll.

(1) Von unten gesehen (Zeile 75-67), gibt es im letzten Block (11) vier Strenge vor dem Verb (Zeile 73-70). Da keines der Strenge mit einer Präposition beginnt, können diese nur dem Objektiv mit dem Hinweiser (p_0) zugeordnet werden (siehe Tab. 2). Die vier Strenge (*das gleiche Denken hier*) bekommen deswegen alle die gleiche Bestimmung, d. h. Code 50.

Tabelle 6
Algorithmische Codierung

Zeile	*Code*	*Streng*	*B*	*Zeile*	*Code*	*Streng*	*B*	*Zeile*	*Code*	*Streng*	*B*
1	00	[.]	1	27	50	eben		53	40	sparen	
2	01	*		28	01	,	4	54	50	*	
3	30	*		29	30	ich		55	01	*	9
4	40	Schauen		30	40	habe		56	30	*	
5	50	Sie		31	50	ja		57	40	könnte	
6	50	sich		32	50	mein		58	50	*	
7	01	die		33	50	Geld		59	01	,	10
8	50	Einstellung		34	01	,	5	60	30	das	
9	60	von		35	01	wieso		61	40	ist	
10	60	heutzutage		36	30	*		62	50	mir	
11	60	an		37	40	soll		63	01	doch	
12	60	*		38	50	ich		64	50(90)	völlig	
13	01	,	2	39	01	denn	6	65	50(90)	egal	
14	01	und		40	30	*		66	00	.	
15	30	es		41	40	mithelfen		67	01	*	11
16	40	ist		42	50	*		68	30	Es	
17	50	ja		43	01(90)	,		69	40	ist	
18	50	nicht		44	50(90)	Wege		70	50	das	
19	50	nur		45	01(60)	zu	7	71	50	gleiche	
20	60	unter		46	30	*		72	50	Denken	
21	60	den		47	40	finden		73	50	hier	
22	60	Gemeinde angestellten		48	50	*		74	00	.	
23	01	,	3	49	01	wie	8	75	01	[*]	
24	01	die		50	01	die					
25	30	meisten		51	30	Gemeinde					
26	40	meinen		52	30	Geld					

(2) Noch einmal, in seiner Aufwärtsbewegung finden die Routinen nur einen Streng nach dem Verb (Zeile 68), was bedeutet, dass Block 11 einen explizit ausgedrückten Agentenstreng beinhaltet. In seiner Bestimmung bekommt er den Code 30 zugeordnet. Auf der Zeile (67) gibt die Satzmarkierung an, dass ein neuer Satz beginnt.

(3) Die algorithmische Codierung setzt nun damit fort, dass der Rechner alle Verben aufsucht. Gleichlaufend mit der Zeile (58) muss er aber auch noch eine Leerzeile zwischen der Satzmarkierung (,) und dem Verb *könnte* bereitstellen, da das Verb in einem solchen Fall keinen markierten Objektivstreng hat. Dieser Streng kann nun eingefügt und angegeben werden. Außerdem kann der eingefügte Streng auch noch mechanisch mit dem richtigen Code versehen werden.

(4) Dieses Verfahren gilt für beide Verben (*könnte*, *sparen*). Für das erste Verb lässt sich der implizite Agent eindeutig ausmachen und kann daher mit (*) für den fehlenden Streng versehen werden. Von unten gesehen ist der Agentenstreng für das zweite Verb (Zeile 53) in der vorgegebenen Prozessordnung explizit ausgesagt.

(5) Aus Tabelle 6 wird weiterhin ersichtlich, dass der Rechner auf seinem Weg nach oben auf Zeile 47 das Verb *finden* feststellt. Eine mechanische Markierung wird sowohl für das Objektiv als auch für den Agenten benötigt.

(6) Danach muss die Leerstelle bestimmt werden, die nach dem Satzmarkör (,) entsteht. Der Streng zwischen dem (,) und dem Verb füllt nun die Leerstelle (Zeile 42) aus und bekommt deswegen den Code (50) zugeteilt.

(7) Das Verb *mithelfen* macht eine besondere Aufmerksamkeit erforderlich. Die unmittelbare Satzgrenze (,) trennt *Wege* durch die vorhergehende Umwandlung der Präposition (*zu*) in eine Satzgrenze ab.

(8) Mittlerweile besagt die Prozessregel aber außerdem, dass alle Strenge zwischen zwei Satzmarkierungen, die kein Verb enthalten mit Code (90) markiert werden müssen, die in einem zweiten Schritt in das Objektiv (*) integriert werden, was dazu führt, dass die 90-codes in 50-codes umgewandelt werden (siehe Block 10).

(9) Danach überschreitet der Rechner auf seinem Weg nach oben (Textbeginn) die Satzgrenze *denn* (Zeile 39) und erreicht auf diese Weise den nächsten Satz, der das Verb *soll* beinhaltet. Hier lässt sich vor der abgrenzenden Satzmarkierung *wieso* ein 50-streng feststellen. Ein explizit ausgedrückter 30-streng fehlt aber auf Zeile 36. Aus diesem Grund arbeitet das Programm wiederum in der beschriebenen Weise.

(10) Oberhalb der Grenze von Block 5 liegt eine Strengsequenz vor, die das Verb *habe* enthält. Beide Seiten sind explizit und der Satz wird durch eine Markierung abgegrenzt.

(11) Das Gleiche gilt für den Satz, der dieser Grenze folgt. Von unten gesehen geschieht nun ein algorithmischer Sprung nach oben zum Satz, der das Verb *ist* (Zeile 16) enthält. Was diesen Satz auszeichnet, ist ein expliziter 30-streng und drei 50-strenge sowie drei 60-strenge, die mit dem Hinweiser (p_1=unter) eingeleitet werden. Nachdem der Mechanismus den Strengen die entsprechenden Codes zugeordnet hat, kann die letzte Grenze überschritten werden.

(12) Wie sich ablesen lässt, ist das Objektiv differenzierbar, d. h. durch Präpositionen, die die Funktion haben, Orientierungspunkte anzuzeigen bzw. auf solche hinzuweisen. Solche Strenge bekommen die Hinweiser Codes (p_1-p_3).

(13) Nach dieser Grenzüberschreitung (Zeile 13) kann schließlich auch der einleitende Satz bearbeitet werden. Eingeleitet wird dieser Satz mit einem Verb, nämlich *Schauen*, was bedeutet, dass der Satz mit einem Platzhalter (*) für den Agenten beginnt. Außerdem muss der Mechanismus auch noch ein Objektiv (*) für die Leerstelle zwischen der Satzgrenze (,) und dem Hinweiser *an* einsetzen.

Damit sind die Operationen abgeschlossen. Das Resultat der algorithmischen Bearbeitung hat durch die Zusammensetzungen der Verben an zwei Stellen zu einer Ausdehnung geführt, die mit technischen Markierungen gekennzeichnet wurden. Außerdem hat der Algorithmus mehrere (*) für implizite Strenge auf beiden Seiten des Verbs eingesetzt. Diese Maßnahme erlaubt es im weiteren Verlauf die Perspektive des Textproduzenten sichtbar zu machen, die sich zwischen den Zeilen verbirgt. Im Prozess müsste der Mechanismus nun alle Strenge erkannt und mit Codes versehen haben.

Übung 5: Codieren von Blöcken und Verschiebung

Die gesamte Lösung lässt sich noch einmal in Tabelle 7 kontrollieren. Nun geht es darum, erkannte Blöcke (B) als Informationsquelle für das Auswechseln von Platzhaltern (*) mit Strengen einzuführen, was in Tabelle 7 demonstriert wird.

Tabelle 7

Codieren und Supplieren von Blöcken

Zeile	*Code*	*Streng*	*B*	*Supplement*	*Zeile*	*Code*	*Streng*	*B*	*Supplement*
1	00	[.]	1		39	01	Denn	6	
2	01	*			40	30	*		B5=[[ich]]
3	30	*		[X]	41	40	mithelfen		
4	40	Schauen			42	50	*		Wege
5	50	Sie			43	01	,		
6	50	sich			44	50	Wege		
7	50	die			45	01	zu	7	
8	50	Einstellung			46	30	*		B6=[[[ich]]]
9	60	von			47	40	finden		
10	60	heutzutage			48	50	*		B8=[[wie die Gemeinde Geld+ wie die Gemeinde Geld+wie die Gemeinde das +mir+doch völlig egal]]
11	60	an			49	01	wie	8	
12	60	*		B2=[es+ja nicht nur +unter den Gemeinde angestellten]	50	01	die		
13	01	,	2		51	30	Gemeinde		
14	01	und			52	30	Geld		
15	30	es			53	40	sparen		
16	40	ist			54	50	*		B9=[wie die Gemeinde Geld+das+mir+doch völlig egal]
17	50	ja			55	01	*	9	
18	50	nicht			56	30	*		B8=[wie die Gemeinde Geld]
19	50	nur			57	40	könnte		
20	60	unter			58	50	*		B10=[das+mir+doch völlig egal]
21	60	den			59	01	,	10	
22	60	Gemeinde angestellten			60	30	das		
23	01	,	3		61	40	ist		
24	30	die			62	50	mir		
25	30	meisten			63	50	doch		
26	40	meinen			64	50	völlig		
27	50	eben			65	50	egal		
28	01	,	4		66	00	.		
29	30	ich			67	01	*	11	
30	40	habe			68	30	Es		
31	50	ja			69	40	ist		
32	50	mein			70	50	das		
33	50	Gehalt			71	50	gleiche		
34	01	,	5		72	50	Denken		
35	01	wieso			73	50	hier		
36	30	*		B4=[ich]	74	00	.		
37	40	soll			75	01	[*]		
38	50	ich							

Kommentar: Wenn eine Satzgrenze zweiten Grades (CM) dem Agenten (*) vorangeht, wird der notwendige Graphemstreng vom vorhergehenden 30-streng abgerufen (=kopiert). Ist es dagegen eine Satzgrenze ersten Grades (SM), werden die notwendigen Graphemstrenge von beiden, den 30-strengen und den dazugehörenden [50/(60, 70, 80)] -strengen abgeholt. Im Falle eines Abschnitts oder dem Beginn eines Textes gibt es keine kopierbare Strenge. Dieses Verhältnis wird mit Variablen (X) vermerkt.

Übung 6: Messenger zur Identifikation von Rotationen

In dieser Phase wird es notwendig, den Text so vorzubereiten, dass Strengrotationen bestimmt und gemessen werden können. Die Messungen geschehen mit Hilfe der Werte, die in Tabelle 8 definiert werden.

Tabelle 8
Empirisch definierte Mitteiler

	Linke Seite vom Satz:	*Rad*[1]		*Rechte Seite vom Satz:*	*Rad*[1]
A	*=vor dem Verb*	*[iϕ/2]*	*O*	*=nach dem Verb*	*[iθ2]*
1	$SM_p+*_A+\omega$	0	1	$\omega+*_O+SM$	0
2	$SM+CM+*_A+\omega$	0.785	2	$\omega+*_O+CM+Phrase+CM$	0.785
3	CM+Phrase+ω	1.57	3	$\omega+Prä+*_O+SM$	1.57
4	CM+Prä+Wort +ω	2.36	4	$\omega+Prä+*_O+...+CM$	2.36
5	Wort+ω	3.14	5	ω+Wort	3.14
6	Wort+Prä+ω	3.87	6	ω+auf_Prä+Wort	3.87
7	Wort+Prä+ …+ω	4.71	7	ω+mit_Prä+Wort	4.71
8	$CM+*_A+\omega$	5.50	8	ω+für_Prä+Wort	5.50
9	$SM_s+*_A+\omega$	6.28	9	$\omega+*_O+CM$	6.28

[1]Die Berechnung mit Radianten (Rad) entspricht: [arc $\alpha = 2\pi(i\phi/360)$] und [arc $\beta = 2\pi(i\theta/360)$]. Hestenes (1986/1993, p. 75) unterstreicht die Notwendigkeit der exponentiellen Funktion und deren Serienexpansion, die aber die Messung durch Radianten erforderlich macht.
SM_p: Beginn eines Abschnittes; SM_s: Beginn eines Satzes;

Die Informationsträger sind hier sogenannte Mitteiler, die mit dem Ausgangspunkt in empirischen Fällen gewonnen wurden. Die angegebenen Mitteiler beziehen sich auf rhythmische Pendelbewegungen und Neigungen, die sich in Übereinstimmung mit dem entdeckten Uhren-Modus (B. Bierschenk, 2001) entwickeln. Pendelbewegungen schaffen sowohl für die (A) Komponente als auch für die (O) Komponente die Voraussetzung für jeden einzelnen Block feststellen zu können, welcher Rotationsfall infrage kommen muss. Eine Strengsequenz entspricht einem Muster, das auf dem ersten Niveau auftritt und damit als textuelles Muster erster Ordnung definiert wurde. Ein Muster, das auf dem zweiten Niveau auftritt, entspricht einem Mitteiler, der als Muster zweiter Ordnung definiert ist. Identifikationen und Zuordnungen von Mitteilern (M) werden in Tabelle 9 demonstriert.

Die Mitteiler haben lokale Eigenschaften, die die Steuer- und Kontrollbedingungen des funktionellen Satzes vermitteln. Wie aus Tabelle 8 ersichtlich wurde, hat jede Seite neun Grundwerte. Jeder Grundwert entspricht einer Pendelbewegung vor beziehungsweise nach dem Verb. In diesem Schritt wird es nun notwendig, für jeden Block den entsprechenden Rotationsfall anzugeben und das Pendel zu gliedern. Es ist ratsam, die Fälle vor dem Verb mit (A1-A9) und nach dem Verb mit (O1-O9) zu bezeichnen.

Tabelle 9
Identifikationen und Zuordnungen von Mitteilern

Zeile	*Code*	*Streng*	*B*	*M*	*Zeile*	*Code*	*Streng*	*B*	*M*
1	00	[.]			39	01	Denn	6	A8
2	01	*	1	A1	40	30	*		
3	30	*			41	40	mithelfen		O2
4	40	Schauen		O5	42	50	*		
5	50	Sie			43	01(90)	,		
6	50	sich			44	50(90)	Wege		
7	01	die			45	01(60)	zu	7	A8
8	50	Einstellung			46	30	*		
9	60	von		O6	47	40	finden		
10	60	heutzutage			48	50	*		O9
11	60	an		O4	49	01	wie	8	A5
12	60	*			50	01	die		
13	01	,	2	A5	51	30	Gemeinde		
14	01	und			52	30	Geld		
15	30	es			53	40	sparen		O9
16	40	ist		O5	54	50	*		
17	50	ja			55	01	*	9	A8
18	50	nicht			56	30	*		
19	50	nur			57	40	könnte		O9
20	60	unter		O6	58	50	*		
21	60	den			59	01	,	10	A5
22	60	Gemeinde angestellten			60	30	das		
23	01	,	3	A5	61	40	ist		O5
24	01	die			62	50	mir		
25	30	meisten			63	01	doch		
26	40	meinen		O5	64	50(90)	völlig		
27	50	eben			65	50(90)	egal		
28	01	,	4	A5	66	00	.		
29	30	ich			67	01	*	11	A5
30	40	habe		O5	68	30	Es		
31	50	ja			69	40	ist		O5
32	50	mein			70	50	das		
33	50	Geld			71	50	gleiche		
34	01	,	5	A8	72	50	Denken		
35	01	wieso			73	50	hier		
36	30	*			74	00	.		
37	40	soll		O5	75	01	[*]		
38	50	ich							

Zum Beispiel signalisiert A1 einen Mitteiler, der voraussetzt, dass unmittelbar nach dem Satzbeginn ein (*) und ein Verb folgt *[SM+*+Verb]*. Im zweiten Block ist der Agentenstreng explizit ausgedrückt, was mit dem Fall A5 übereinstimmt, nämlich ein oder mehrere 30-strenge haben das Verb als ihre untere Grenze *[CM+Wort+Verb]*. Auf der Objektiv-Seite sind im ersten und zweiten Block die entsprechenden 50-strenge mit dem O5-Fall angegeben *[Verb+Wort+CM]*. Es gibt aber auch noch einen O4-Fall (Zeile 11) im ersten Block [*Präposition+*+CM]* und einen O6-Fall (Zeile 20) im zweiten Block [*60-preposition+Wort+CM*]. Im ersten Block sucht der Algorithmus deswegen nach einem

Muster, das mit einem Verb + Präposition beginnt und mit einem (*) vor der Satzmarkierung abschließt. Dieser Messenger stimmt mit dem O4-Fall überein.

Im weiteren Verlauf wird das Verb routinemäßig dem Objektiv zugeordnet. Damit bekommt auch das Verb seinen ausdrücklichen Rotationswert algorithmisch zugeteilt. Die ganze Operation endet mit der Einführung des Mitteilers gegenüber dem Verb. Schließlich steht noch die mechanische Bearbeitung vom O6-Fall aus, der im zweiten Block vorkommt. Hier wird der Prozess durch den (p_1) Hinweiser äunteró gesteuert. Auf diese Weise setzt sich der Zuordnungsprozess fort.

Im dritten und vierten Block (Zeile 23-33) stimmen beide Seiten des Verbs mit Fall A5 + O5 überein. In Block fünf wird es dagegen notwendig, dass der Agentenstreng von Block vier heruntergeholt wird, was mit der Bedingung für das Muster A8 in Einklang steht. Dieses Beispiel unterscheidet sich von A1 dadurch, dass sein Rotationswert durch die Fortschreibung des vorhergehenden Blocks bestimmt wird. Ein ähnlicher Prozess wird auch in Block 6 (Zeile 39) für die Komponente (A) angezeigt.

Im weiteren Verlauf verändert sich nun noch einmal das Verhältnis auf der Seite für (O). Mit dem Fall O9 ist eine Strengsequenz verbunden, die den Satz nach unten öffnet und damit eine komplexe Integration von Strengen einleitet, die diese Leerstelle ausfüllen. Was aus der Tiefe geholt wird, ist Block 10, der auf seinem Weg nach oben die Blöcke 9 und 8 mitnimmt. Schließlich ist der letzte Block durch A5 + O5 gekennzeichnet.

Wie bereits in der Codierung von impliziten funktionellen Sätzen deutlich wurde, bezieht sich der Fall O2 *[Verb+* +CM+Phrase+SM]* auf die Formel, die besagt, dass alle verblose Strenge (= Phrase) mit Code (90) markiert werden müssen, aber in einem zweiten Schritt in das Objektiv (*) integriert werden, d. h. in Code (50) umgewandelt.

Kalkulieren

Übung 7: Strengrotation

Da die Rotationen ein Maß für strukturelle Texttiefen darstellen, wird die Berechnung der Strengrotationen dreidimensional beschrieben. Tabelle 10 stellt die prismatischen Eigenschaften rotierender Strenge vor.

Tabelle 10

Superstreng-Windungen

Muster	*Eigenschaft*	*Windungsgröße*
Messenger	Virtuell	Grundwert (W=1/1)
+Word	Physisch	Wickelwert (W=1/10)
+Graphem	Materiell	Klappenwert (W=1/100)

Die nun folgenden drei Annahmen wurden durch experimentelle Studien validiert (B. Bierschenk, 2001; I. Bierschenk & B. Bierschenk, 2004).

(1) Da ein Mitteiler nur virtuell existiert, bezieht sich die Berechnung auf den Basiswert jeder einzelnen Komponente. Ihr Windung hat den Grundwert (W=1/1).

(2) Da aber die Muster in einem Text auch auf einem physischen Niveau existieren, werden die Variablen auf diesem Niveau in der Form von Worten mit der Windung von (W=1/10) berechnet, was bedeutet, dass vorausgesetzt wird, dass eine Komponente in der Regel aus nicht mehr als 10 Worten besteht. Dieser Wert wird zum Grundwert addiert.

(3) Auf dem realen Niveau konzentriert sich die Berechnung auf die Grapheme und deren mitschwingenden Eigenschaften, was mit der Windung (W=1/100) zum Ausdruck kommt und mit der Anzahl der realisierten Grapheme multipliziert wird. Auch hier gilt die empirische Beobachtung, dass ein Wort in der Regel aus nicht mehr als zehn Graphemen besteht.

Wie die Rotationsgeschwindigkeit und Akzeleration einzelner Strenge berechnet wird, soll nun mit einem Beispiel aus Tabelle 11 verdeutlicht werden.

Tabelle 11
Intervallbasierte Strengrotationen

Zeile	*Code*	*Streng*	*B*	*P*	*Summe*	*Zeile*	*Code*	*Streng*	*B*	*P*	*Summe*
1	00	[.]				39	01	denn	6	A8	
2	01	*	1	A1		40	30	*			1.753844
3	30	*			0	41	40	mithelfen		O2	
4	40	Schauen		O5		42	50	*			1.1304
5	50	Sie				43	01	,			
6	50	sich				44	50	Wege			
7	50	die				45	01(60)	zu	7		
8	50	Einstellung			5.5892	46	30	*		A8	-0.86015
9	60	von		O6		47	40	finden		O9	
10	60	heutzutage			5.1471	48	50	*			-9.60956
11	60	an		O4		49	01	wie	8	A5	
12	60	*			-4.08791	50	01	die			
13	01	,	2	A5		51	30	Gemeinde			
14	01	und				52	30	Geld			4.9612
15	30	es			4.2704	53	40	sparen		O9	
16	40	ist		O5		54	50	*			-1.64499
17	50	ja				55	01	*	9	A8	
18	50	nicht				56	30	*			3.272625
19	50	nur			4.8042	57	40	könnte		O9	
20	60	unter		O6		58	50	*			2.927591
21	60	den				59	01	,	10	A5	
22	60	Gemeinde angestellten			6.1146	60	30	das			3.8936
23	01	,	3	A5		61	40	ist		O5	
24	01	die				62	50	mir			
25	30	meisten			4.4274	63	50	doch			
26	40	meinen		O5		64	50	völlig			
27	50	eben			4.082	65	50	egal			
28	01	,	4	A5		66	00	.			5.6834
29	30	ich			3.8936	67	01	*	11	A5	
30	40	habe		O5		68	30	Es			3.5168
31	50	ja				69	40	ist		O5	
32	50	mein				70	50	das			
33	50	Gehalt			4.8984	71	50	gleiche			
34	01	,	5	A8		72	50	Denken			
35	01	wieso				73	50	hier			
36	30	*			5.046333	74	00	.			5.7776
37	40	soll		O5		75	01	[*]			
38	50	ich			3.9878						

Der Algorithmus beginnt wieder von unten und bewegt sich in seinem Arbeitsgang nach oben. Daher ist der letzte Satz aus Tabelle 9 erneuter Ausgangspunkt. Von unten gesehen beginnt die Messung mit der Berechnung der Bogenmaße (Rad) für jeden einzelnen O5-streng, was der Tatsache entspricht, dass die 50-strenge Teile einer Serie im O5-Fall sind. Als Resultat der Berechnung erhält die O-Komponente den Wert (3.14/10), d. h. (Rad=0.314). Zu diesem Wert addiert der Rechner einen weiteren Teil, nämlich die Anzahl (N) der Grapheme, die die Komponente aufbaut.

Diese Operationen erlauben es dem Algorithmus auch Schwingungen der Grapheme zu berücksichtigen, d. h. die O-Summe, multipliziert mit der Anzahl der Grapheme ((0.314/10) *N). Schließlich muss auch noch der Grundwert von O5 hinzugerechnet, d. h. (=3.14). Die gesamte Serie summiert sich zum Wert (Summe für O = 5.7776). Auf die gleiche Weise summiert sich in Block 11 der Wert für den Agenten zur (Summe für A = 3.5168).

In Tabelle 9 tauchen aber auch noch besondere Umstände auf, die den Rechner in die Lage versetzen, mit impliziten Strengen rechnen zu müssen. Wie bereits beschrieben, wird der Wert (Code 30) des vorangegangenen Blockes (8) heruntergeholt. Indessen zeigt das Ziehen der Wurzel aus (Rad=4.9612) an, dass es sich hier um *Schatten* handelt (B. Bierschenk, 2001, S. 11). Da der entsprechende Textausschnitt aber mit einem Komma endet (Zeile 59), dem außerdem sowohl ein Agentenstreng als auch ein Objektivstreng folgt, wird auch der Fall (O9) aktiviert. Für diesen Fall muss der Rechner die Wurzeln aus den in Block 10 explizit ausgedrückten Komponenten (A5) und (O5) ziehen.

Wenn die in Tabelle 11 dargestellten Messungen der Pendelbewegungen ausgeführt sind, erscheinen Unterschiede zwischen den Blöcken, zum Beispiel zwischen Block 9 und anderen Textstellen, die unterschiedliche Schatten produzieren. Wenn also funktionelle Sätze Löcher im Text hinterlassen, bilden sich Schatten, die mehr oder weniger große Tiefen signalisieren.

Übung 8: Strangrotation

Wenn alle Bedingungen für die Messung der Rotationen durch Radianten erfüllt sind, kann der Algorithmus damit beginnen, die Vorbereitungen für eine dreidimensionale geometrische Präsentation einzuleiten. Wie die dafür notwendige Tabelle für die Summenvariablen aufgestellt werden muss, wird in Tabelle 12 illustriert.

Tabelle 12
Intervallbasierte Strangrotationen

Paar	*Rad*	*Rad*	*Intervall*	*Fall*
Nummer	*α-Strang*	*β-Strang*	*Nummer*	*Nummer*
1	0	5.5892	1	1
2	0	5.1471	1	2
3	0	-4.08791	1	3
4	4.2704	4.8042	2	1
5	4.2704	6.1146	2	2
6	4.4274	4.082	3	1
7	3.8936	4.8984	4	1
8	5.046333	3.9878	5	1
9	1.753844	1.1304	5	2
10	-0.86015	-9.60956	5	3
11	4.9612	-1.64499	5	4
12	3.272625	2.927591	5	5
13	3.8936	5.6834	6	1
14	3.5168	5.7776	7	1

Wie aus der Aufstellung hervorgeht, ist es ratsam die Summen der Paare (A=α) und (O=β) zu parallelisieren, weil das die rechnerische Arbeit erleichtert. Für jede Komponente sind weiterhin folgende drei Parameter angegeben: (1) Intervall, (2) Fall (1, 2, 3, é) je Intervall und (3) die Rotationen (in Rad) der (α) und (β) Variablen. Demzufolge werden fünf Kolonnen benötigt. Um für den gesamten Beispielstext die Summen für die Paare (1-14) in eine geometrische Illustration (siehe Abb. 1) umwandeln zu können, muss der Algorithmus die Werte der Strangrotationen in ein Programm einbringen können, das Graphen konstituiert.

Kommentar: Die zentrale Einheit ist weiterhin die (A/O) Einheit. Deswegen werden in der ersten Kolonne alle Paare angeben. Da es mehr als eine (O) Variable in einem einzelnen Satz geben kann, werden gleichzeitig aber auch alle unterschiedlichen Variablen vermerkt. Aus diesem Grunde sind die Paare in der ersten Kolonne 14 in ihrer Anzahl anstelle der in Tabelle 7 angegebenen Anzahl der Blöcke (B=11).

Geometrische Repräsentation

Nun wird es notwendig, die Tabellenwerte (Tabelle 12, Kolonne 5) so zu transformieren, dass die Werte in ein Graphenprogramm (z. B. SigmaPlot, 2008, Version 11) eingegeben werden können. Nachdem alle (XYZ)-Werte in der verlangten Ordnung eingelesen wurden, wird es erforderlich, den Typ der Graphen anzugeben, der generiert werden soll. Die Angaben sollten sich also hierbei auf die geometrische Darstellung eines 3D-Gitters beziehen. Das Resultat der generierten Graphen wird unten in Abbildung 1 zugänglich. Wie sich die Texträume nach der Datenbearbeitung gestaltet haben, wird sowohl in den Graphen für die Intention als auch für die Orientierung ersichtlich.

Übung 9: Entfaltung der Texträume

Eine erste Bemerkung zum Aussehen der entfalteten Texträume hat mit dem Einlesen der Daten zu tun. Wie beim gewöhnlichen Lesen werden diese von links nach rechts dargestellt. So zu arbeiten ist einfach am natürlichsten. Damit lässt sich die Textprogression im jeweiligen Graphen von rechts nach links ablesen. Wie sich die Flussdynamik des transformierten Textbeispiels entwickelt, geht aus Abbildung 1 hervor.

Intention Ihre Texturen treten wie Stücke von wogenden Tüchern in Erscheinung. Die Entfaltung in (A) beginnt im ersten Intervall mit dem Wert (0.00). Danach steigt die Textur über die Intervalle (2, 3, 4) an, was zu einem gleichförmigen Höhenrücken führt. Ein Sattel formt sich im vierten und fünften Intervall. Im Übergang vom fünften zum sechsten Intervall bildet sich in (A) ein trichterförmiger Abschnitt mit einem Tief (~-0.86), das aber dann von einem massiven Höhenrücken mit Werten, die zwischen (~+3.51) und (~+4.96) liegen, abschließt. (A) beginnt also mit einem genuinen Nullpunkt, der sich wie ein Anker verhält. Das wogende Tuchstück markiert dementsprechend nicht so sehr die Tiefe, sondern macht vielmehr deutlich, dass die Entwicklung auf *substanzielle* Abschnitte ausgerichtet ist.

Orientierung Was sich durch das Tuchstück auf der rechten Seite abbildet, beginnt im ersten Intervall mit einem Wert (~+5.58). Danach geht die Textur bis zum fünften Intervall in eine ziemlich gleichförmige Entwicklung über, die aber danach abtaucht und ihren niedrigsten Punkt in (~-9.60) erreicht. Das Resultat ist ein dunkler und trichterförmiger Abschnitt. In den beiden letzten Intervallen bestimmen die Werte (~+5.68 und ~+5.77) die Erscheinung und bilden damit die linke Kante eines schalenförmigen Trichters.

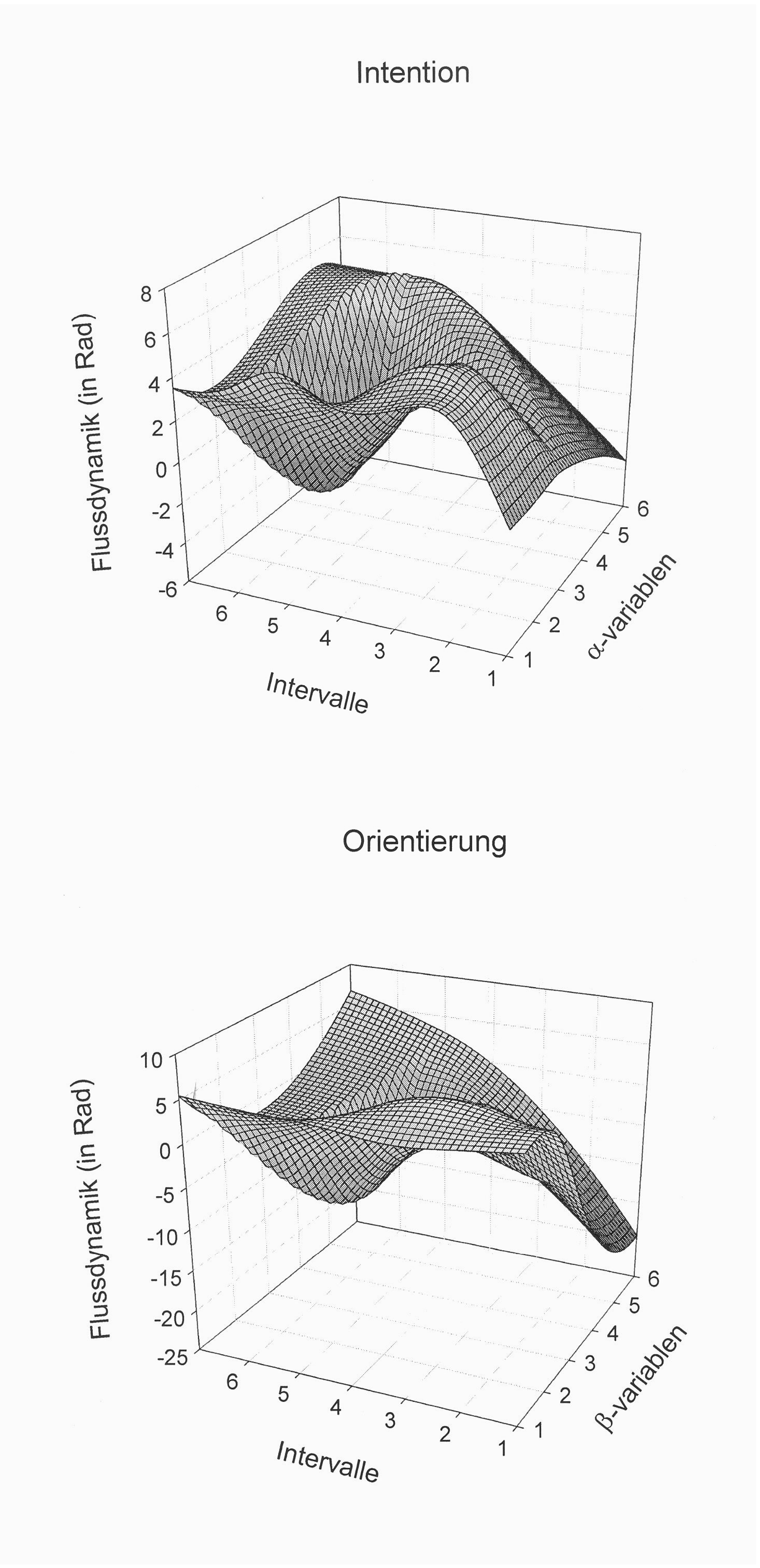


Abbildung 1 *Entfaltete Texträume*

Kommentar zu (A) und (O): Wo (O) eine leicht wogende Textur hat, weist (A) einen relativ steilen Anstieg und die Form eines Massivs auf. Wo (O) die größte Tiefe erfasst, macht sich die komplementäre Substanz in (A) am deutlichsten bemerkbar. Was also als Substanz hervortritt, steht im Fokus des Textproduzenten, d. h., ist im Zentrum seiner Perspektive. Bei einer genaueren Analyse der Graphen lässt sich eine übergreifende Symmetrie beobachten.

Was bisher beobachtet werden konnte, ist die Entfaltung einer Dynamik, die den Textfluss kennzeichnet. Bisher ist gezeigt worden, wie die offenen Sphären von Texten durch die Kopierfunktion kontrolliert und ergänzt werden. Was dabei beobachtet werden konnte, ist die Tatsache, dass die Kopierfunktion in Richtung auf wachsende Strangvektoren hin arbeitet. Im Hinblick auf die entfalteten Räume ist es deutlich so, dass visköse und elastische Texteigenschaften die Rotationsgeschwindigkeit beeinflussen und auf diese Weise die Stressfunktionen im Textraum beherrschen. Auf diesem Hintergrund kann nochmals festgestellt werden, dass die Kopierfunktion rollende Vektoren kontrolliert, die in den Texträumen als wogende Tuchstücke hervorgetreten sind.

Eine zentrale Aufgabe der Kopierfunktion besteht weiterhin darin, vollständige [AaO] Einheiten aus unvollständigen, d. h., deformierten Einheiten, herzustellen. Die Effekte, die durch Deformationen entstehen, werden in der Brechung von mehrfach geschichteten und gewickelten Variablen erkennbar. Wachsende Strangvektoren können mit rotierenden Akzelerationen gleichgestellt werden, die das Resultat der Unterschiede zwischen der Energie sind, die in vollständige [AaO] investiert wurde und jener, die in unvollständige Einheiten gepresst wurde.

Eine ganz andere Frage stellt sich, wenn es darum geht, die Energie zu veranschaulichen, die in eine Strangproduktion investiert wurde. Diese Art der Unterschiede erscheint also an der Textoberfläche als wogende Textur. Dass die innere Dynamik Einfluss auf die Texture nimmt, fordert eine nähere Untersuchung der Eigenschaften die sich in der Texttiefe verbergen.

Übung 10: Intervallbezogene Alpha und Beta Ordnungen

Die Fragestellung, die nun weiter verfolgt werden soll, gilt deshalb der Messung der inneren Dynamik. Um die Konzentration von konservierten Energien und die damit verbundenen Radianten entdecken zu können, muss ein Prinzip für die Gruppierung der Alpha (α) und Beta (β) Variablen eingeführt werden können, was bedeutet, dass die Variablen Eigenschaften anderer Art reflektieren werden als jene, die die Rotationen veranlassten. Offensichtlich handelt es sich hierbei um eine Konzentration von *genützten Energien*, die sich auf eine Funktion gründet, die an bestimmten Stellen die Faltung von produzierten Strängen steuert.

Fadenfaltung Um dem Ursprung der Fadenfaltung deutlich machen zu können, ist es ratsam, den Algorithmus für sowohl (A) also auch (O) eine Tabelle generieren zu lassen, die aus zwei Kolonnen besteht, wie es in Tabelle 13 und Tabelle 14 gezeigt wird.

Um das erreichen zu können, muss der Rechner die Werte verwenden, die in den Intervallen der Tabelle 12 enthalten sind. Der Rechner muss z. B. für jede Variable von (O) eine Zeile generieren und gleichzeitig die Markierung abbuchen. Wenn er fertig ist, hat er eine Kolonne für die β-Variablen und eine für die Rotationswerte (Rad) des β-Strangs aufgestellt, was in Tabelle 13 veranschaulicht wird. Connes (1994) hat eine Operandenfunktion (Δ) vorgeschlagen, die sich auf die binäre Gruppe (G*) bezieht. Es hat sich nämlich gezeigt, dass diese Funktion zu bemerkenswert einfache Messungen zur Faltungen, Verzweigungen und der Entwicklung von Baumstrukturen führt.

Tabelle 13
Gruppierung in der O-Komponente

Variable	*Radian*
Nummer	*β-strang*
.	
1	5.5892
2	5.1471
3	
,	
4	4.8042
5	6.1146
,	
6	4.082
,	
7	4.8984
,	
8	3.9878
10	
11	
12	
,	
,	
13	2.97591
.	
14	5.6834
,	
15	5.7776
.	
9	1.1304
,	
12	-1.64499
,	
3	-4.08791
,	
11	-9.60956

Tabelle 14
Gruppierung in der A-Komponente

Variable	*Radian*
Nummer	*α-strang*
.	
1	
2	
3	
,	
4	4.2704
5	4.2704
,	
6	4.2704
,	
7	3.8936
,	
8	5.046333
9	
,	
10	
11	4.9612
,	
12	3.272625
,	
13	3.8936
,	
14	3.5168
.	
9	1.753844
,	
1	0.00
2	0.00
3	0.00
,	
10	-0.86015

Im koordinativen Zusammenspiel der Variablen nimmt die Δ-Funktion auch auf sehr kleine Unterschiede Rücksicht, wenn es darum geht, die Entwicklung und das Erscheinen einer Struktur zu verdeutlichen. Im Faltungsprozess ist der numerische Wert der Δ-Funktion immer die Hälfte der Unterschiede zwischen dem Effekt einer Referenzvariablen und dem operativen Wert, der Variablen die am Fusionsprozess teilnehmen. Wenn der Funktionswert (Δ/2) den Ausgangspunkt für eine Fusion bildet, werden die Variablen vereinigt, die zu diesem Zeitpunkt operieren. Die Vereinigung gründet sich auf eine 2⊗2-Matrix. Der operierende Wert wird in dieser vierzelligen Matrix dabei in die obere linke Zelle eingegeben, währende

der kontrastierend Referenzwert in die untere rechte Zelle eingeführt wird. Alle übrigen Zellen werden danach mit Null-Werten gefüllt. Auf diese Weise identifiziert der *Reisverschluss* (B. Bierschenk, 2002) diese Zellen mit dem Alter Ego des jeweiligen Wertes.

Die beschriebene Arbeitsweise bedeutet, dass die Variablenwerte durch Operationen eingebettet werden, die alle offenen Plätze schließen. Diese Verfahrensweise lehnt sich an Heisenbergs Unsicherheitsprinzip an (Greene, 1999, S. 424), und ist, gemäß Mackenzie (1997) ausreichend, um einen Raum zu generieren, für den das gesamte Standardmodell der elementaren Partikelinteraktion Gültigkeit hat. Auf diesem Hintergrund sind die *Attraktoren* der Faltungsräume das Resultat der Relation [$T=C\otimes C$]. Die progressive Spur [T] der angegebenen Relation schafft damit die Voraussetzung für ein erfolgreiches Zusammenflechten der Variablen zu komplizierten Strukturen und einen natürlichen Ausdruck für *funktionelle Identität.* Damit kann also festgestellt werden, dass sich der Konzentrationsraum auf der Basis von zwei einfachen Kopplungsmatrizen (C) entwickeln muss.

Kommentar: Der von B. Bierschenk (2002) eingeführte Mechanismus baut also auf der Assoziation von zwei diskreten Punkten auf. Da nun der Faltungsprozess von nur zwei Punkten und deren Alter Ego ausgeht, können klassische arithmetische Operationen ausgeführt werden trotz der Tatsache, dass jeder Punkt unauflöslich mit seinem Alter Ego verflochten ist.

Nachdem auch alle Vergleiche für Tabelle 14 durchgeführt sind, müsste der Fusionsbaum eine Ordnung aufweisen, die das Resultat der (A) Gruppierung widerspiegelt. Wie bereits erwähnt, ist die Bildung von binären Gruppen der Ausgangspunkt für die Faltung. Mit dieser Voraussetzung als Grund für die Erstellung von Fusionsbäumen vergleicht der Rechner innerhalb eines Intervalls immer nur zwei Rotationswerte. Alle Differenzen müssen sich im kritischen Intervall ($0>\omega<1$) bewegen. Für den Aufbau des Fusionsbaums ist es nun notwendig, die entstandenen Differenzen so zu nutzen, dass zeitabhängige Gruppierungen entstehen. Diese Gruppierung berücksichtigt also den Zeitaspekt, d. h. die Nachbarschaft der Variablen innerhalb der gegebenen Intervalle.

Bei der Berechnung dieser Fusionsdynamik werden außerdem alle Werte untersucht, die sich innerhalb eines Intervalls befinden. Alle benachbarten Werte müssen dabei miteinander verglichen werden, um danach Mittelwerte bilden zu können. Sollte sich der Mittelwert eines Paars im kritischen Intervall bewegen, liegt die Voraussetzung vor, den Mittelwert für (G*) weiteren Vergleichen zugrunde zu legen. Sobald aber ein (G*) Wert außerhalb des kritischen Intervalls liegt (z. B. Variable 11, in Tab. 12), isoliert der Rechner diesen Wert und nimmt ihn aus der gegebenen Ordnung heraus. Der auf diese Weise isolierte Wert wird für seine spätere Anwendung am Ende der Tabelle wieder eingeführt. Für jede isolierte Variable wird am Ende einer Tabelle die entsprechende Zeile generiert. Erst wenn der Algorithmus mit einem Intervall fertig ist, kann die gegebene Intervallgrenze überschritten werden. Auf diese Weise wird im beschriebenen Prozess Universalität eliminiert, die ohnehin mit einer evolutionären Theorie unvereinbar ist.

Übung 11: Dimensionierung zeitabhängiger Netzsysteme

Die Dimensionierung zeitabhängiger Netzsysteme beginnt damit, dass die in den einzelnen Intervallen geschaffenen Differenzen so angewendet, dass ein topologisch bestimmter Fusionsprozess entsteht. Damit ist gemeint, dass die Konvention, die der Erstellung von zeitabhängigen Netzen zugrunde liegt, sich auf die Konstruktion von Koordinaten bezieht, d. h., die Rhomben-förmige (45-gradige) Neigungswinkel haben. Eine

solche Konstruktion lässt sich am besten anhand der Netzbeispiele in den Abbildung 2 und 3 beschreiben.

Der entfaltete Beispielstext weist eine Textur auf, die durch eine Reihe von unbeeinflussbaren und zusammenwirkenden Parametern zustande gekommen ist. Es versteht sich von selbst, dass diese Umstände natürlich kein Bild über die inneren Eigenschaften einer Textproduktion widerspiegeln können, nämlich das, was seine Dynamik verursacht hat. Die Ursachen dynamischer Eigenschaften können leicht mit *Stress* umschrieben werden.

Dafür muss ein Programm aktiviert werden, das im Aufbau von Koordinaten behilflich sein kann. Weiterhin müssen die Voraussetzungen für die Bestimmung der Dimensionen eines entstehenden Netzes geschaffen sein. Der Aufbau beginnt mit der linken Seite, die den Kantenwert (0) bekommt. Auf die gleiche Weise erhält die obere Seite den Kantenwert (0). Auf der linken Seite lässt sich dann folgerichtig die erste Position mit (0:0) und die nächstfolgenden Positionen mit (0:1; 0:2; 0:3; 0:4) angeben. Diese Linksï Rechtsï Dynamik setzt die Ergänzung mit weiteren Dimensionen voraus, was bedeutet das die Dimensionswerte an der oberen Kante die Bezeichnungen (0:0; 1:0; 2:0; é) zugeteilt bekommen.

Die Verknüpfung von zwei Werten signalisiert immer eine Fusion, die in Zustandsattraktoren zum Ausdruck kommt und auf den Koordinatenkreuzen aufzutragen ist. Wie viele Zellen oder Maschen ein Netz beinhaltet, kommt auf die Anzahl der Variablen und Platzhalter (D) an. Von den Plätzen eines Netzes können aber nicht mehr als 74% gefüllt sein (Wales, 2003, p. 12). Diese Relationen sind für unregelmäßige Netze prototypisch, die auf zwei *Stress*-Funktionen aufbauen, nämlich die (X) Funktion, die Gleiten manifestiert und die (Y) Funktion, die Strecken darstellt.

Bis auf weiteres reflektiert die (Z) Funktion gleichseitige Abstände. Da sowohl die Anzahl der Positionen der Attraktoren als auch die der Variablen (=Punktattraktoren) und Platzhalter die Bedingungen für eine ringförmige Anordnung setzen, kann ein ungefährlicher Wert für den Umfang eines Netzes abgemessen werden. Um nun im weiteren Verlauf Unterschiede zwischen den Punktattraktoren, d. h. den terminalen Plätzen an den Kanten und den Zustandsattraktoren in einem Netz vergleichen zu können, werden immer nur die Mittelwerte von binär gebildeten Attraktoren verglichen.

Im Bildungsprozess werden die Werte mit Knoten vereinigt und vorwärts summiert. Die entstandenen Attraktoren müssen dagegen aber rückwärts gebunden werden. Dadurch bekommt ein Terminus seine Kopplung. Mit gebeugten Linien werden gleichförmige Abstände dargestellt und an die Terminalzustände (Punktattraktoren und Platzhalter) gebunden. Ausgehend den Knoten müssen die gebeugten Linien in den Abbildungen 2 und 3 einen Schwalbenschwanz bilden.

Als Resultat verschmelzen sich die Transformationen zu immer komplexeren Zuständen. Auf diese Weise wird zum Beispiel im Koordinatenkreuz (1:1) die räumliche und zeitliche Nähe der Punkt- und Zustandsattraktoren berücksichtigt. Das reicht aber nicht aus. Im Fusionsprozess muss auch noch die Raumverteilung für die Knotenmarkierungen innerhalb eines Netzes einkalkuliert werden können. Deswegen sind die Grenzen so anzugeben, dass die folgenden drei Prinzipien für die Herstellung berücksichtigt werden können:

(1) Keine Linie darf ein und dieselbe Zelle zweimal kreuzen.
(2) Keine Linie darf sich selbst kreuzen.
(3) Der entstehende Gleitfaden muss einen Ring approximieren können.

Nach erfolgreicher Konstruktion müssten nun für den gesamten Text zwei separate Netzsysteme zur Verfügung stehen, die die Unterlage für die Darstellung der Fusionsdynamik durch Energielandschaften abgeben können.

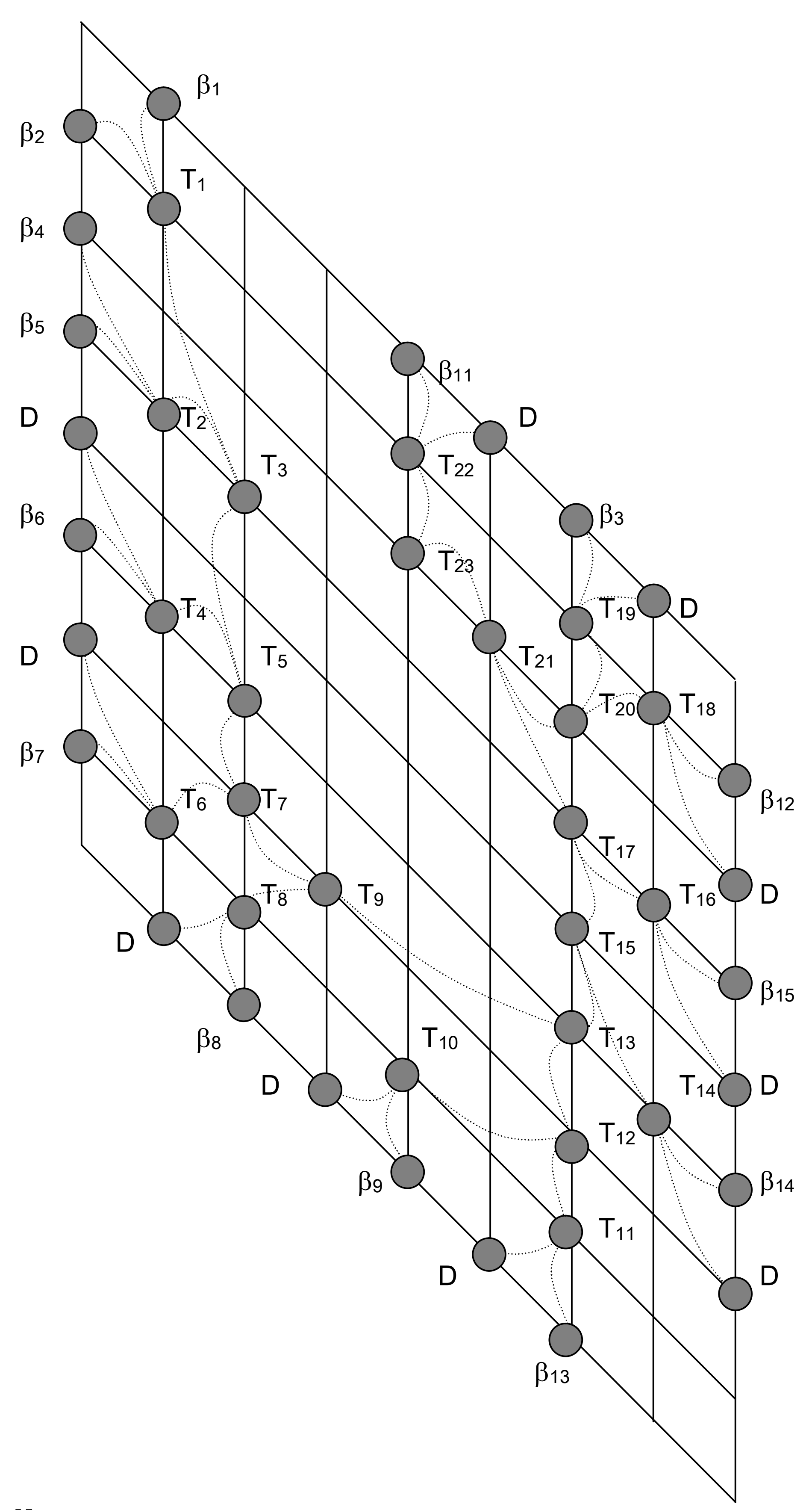


Abbildung 2 ***O**-Netz*

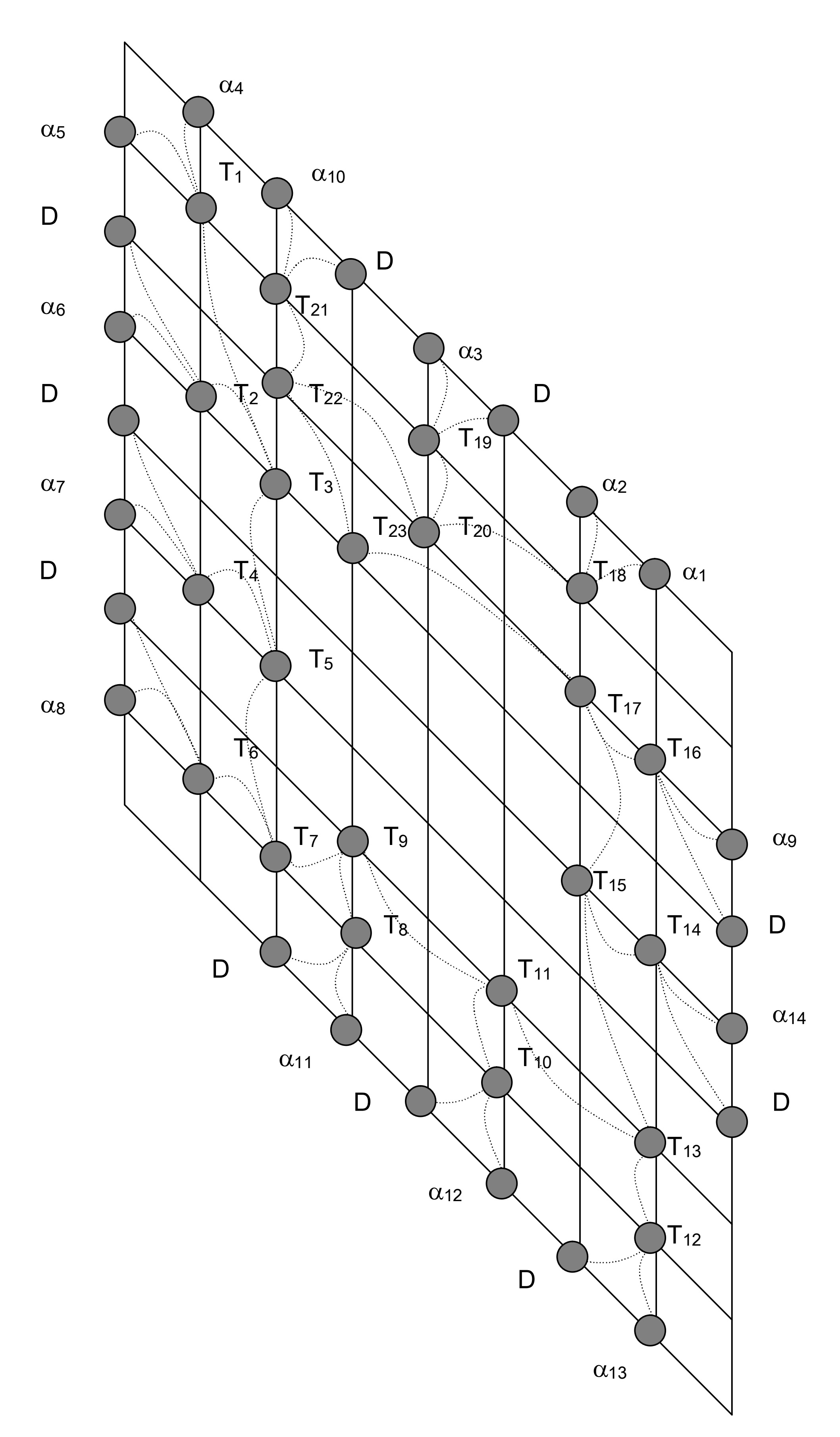


Abbildung 3 ***A**-Netz*

Übung 12: Umwandlung der Koordinatensysteme zu Tabellen

Konnten diese Formation nicht generiert werden, muss irgendwo im Aufbau einen Fehler unterlaufen sein. Nach der erfolgreichen Herstellung von den Netzen geht es in dieser Übung darum, die entstandenen Gitter in Tabellen umzuwandeln. Die notwendigen Hilfsmittel dafür sind eine Kolonne für die Koordinaten (K) und eine für die empirischen Werte (Rad). Die Aufstellung wird in den Tabellen 15 und 16 illustriert.

Tabelle 15

Koordinate für die ***O****-Komponente*

K	*Rad*	*K*	*Rad*	*K*	*Rad*	*K*	*Rad*	*K*	*Rad*
00	0	10	5.5892	20	0	30	0	40	-9.60956
01	5.1471	11	10.7363	21	0	31	0	41	-9.60956
02	4.8042	12	0	22	0	32	0	42	34.84815
03	6.1146	13	10.9188	23	21.6551	33	0	43	0
04	0	14	0	24	0	34	0	44	0
05	4.082	15	4.082	25	25.7371	35	0	45	0
06	0	16	0	26	30.6355	36	34.6233	46	0
07	4.8984	17	4.8984	27	3.9878	37	0	47	1.1304
08	0	18	0	28	3.9878	38	0	48	1.1304

K	*Rad*	*K*	*Rad*	*K*	*Rad*	*K*	*Rad*
50	0	60	-4.08791	70	0	80	0
51	0	61	-4.08791	71	-1.64499	81	-1.64499
52	44.45771	62	-5.7329	72	0	82	0
53	0	63	50.19061	73	5.7776	83	5.7776
54	0	64	44.41301	74	0	84	0
55	0	65	38.72961	75	5.6834	85	5.6834
56	0	66	4.10631	76	0	86	0
57	0	67	2.97591	77	0	87	0
58	0	68	2.97591	78	0	88	0

Tabelle 16

Koordinate für die ***A****-Komponente*

K	*Rad*	*K*	*Rad*	*K*	*Rad*	*K*	*Rad*	*K*	*Rad*
00	0	10	4.2704	20	-0.86015	30	0	40	0
01	4.2704	11	8.5408	21	-0.86015	31	0	41	0
02	0	12	0	22	-0.86015	32	0	42	0
03	4.4274	13	4.4274	23	12.9682	33	38.444152	43	0
04	0	14	0	24	0	34	0	44	0
05	3.8936	15	3.8936	25	16.8599	35	0	45	0
06	0	16	0	26	0	36	26.867433	46	0
07	5.046333	17	5.046333	27	21.906233	37	4.9612	47	0
08	0	18	0	28	0	38	4.9612	48	0

K	*Rad*	*K*	*Rad*	*K*	*Rad*	*K*	*Rad*
50	0	60	0	70	0	80	0
51	0	61	0	71	0	81	0
52	0	62	39.304302	72	1.753844	82	1.753844
53	0	63	0	73	0	83	0
54	0	64	37.550458	74	3.5168	84	3.5168
55	0	65	0	75	0	85	0
56	30.140058	66	0	76	34.033658	86	0
57	3.272625	67	0	77	3.8936	87	0
58	3.272625	68	0	78	3.8936	88	0

Nun wird es noch einmal notwendig, die Tabellenwerte so zu transformieren, dass die Werte in ein Graphenprogramm (z. B. SigmaPlot, 2008, Version 11) eingegeben werden können. Dieser Umstand ist für eine 3D-Darstellung der entstandenen Gitter bedeutungsvoll.
Nachdem alle (XYZ)-Werte in der verlangten Ordnung eingelesen wurden, wird es erforderlich, den Typ der Graphen anzugeben, der generiert werden soll. Die Angaben sollten sich also hierbei auf die geometrische Darstellung eines 3D-Gitters beziehen. Das Resultat der generierten Graphen wird unten in Abbildung 4 zugänglich.

Übung 13: Energielandschaften

Struktur verbirgt sich unter einer Textfläche, weshalb es darum geht, Faltungen zu untersuchen, und die damit verbundenen Konzentrationen investierter Energie in Erscheinung treten zu lassen. Wenn sich Faltungen gestalten, erscheinen die Konzentrationen produzierter Energien in gewickelten Texträumen. Verglichen mit den Texturen entfalteter Texträume haben gefaltete Texträume ein anderes Aussehen. Sie erscheinen als gröber gewickelt und markieren, wo symptomatische Hinweise zu finden sind. Perspektivisch können also die Ergebnisse der gefalteten Texträume als *freie Energielandschaften* (Wales, 2003) und *frei von semantischen Überlegungen* begriffen werden. Die neuen 3D-Graphen zeigen, was sich strukturell unter der Oberfläche abzuspielen scheint.

Zum Beispiel kann die Beschreibung einer Landschaft ihren Ausgang darin nehmen, wie sich Höhen formen, Bergformationen hervortreten oder Täler abzeichnen, d. h. mit dem Ausgangspunkt in einer naturgeografischen Betrachtungsweise. Damit können gefaltete Texträume unterschiedlich dargestellt und beschrieben werden. Mit einer solchen Einstellung könnte sich eine einfache Charakteristik für die Landschaften wie folgt gestalten:

Intention Die Landschaft beinhaltet fünf ungefähr gleich hohe Spitzen. Im Übrigen sind große Teile der Landschaft dunkel markiert, was auf eine tief liegende Ebene hinweist.

Orientierung Die Landschaft lässt auf der linken Seite zwei Bergkämme erscheinen, die eine gleichfalls rasch ansteigende kegelförmige Bergformation einrahmen. Im Vordergrund lassen sich zwei Senkungen ausmachen, die beträchtlich unter dem Meeresspiegel (0-Punkt) liegt. Auf der rechten Seite wird ein Vorgebirge sichtbar, das ein paar markante Erhöhungen beinhaltet. Auf dieser Seite zeigt diese Landschaft auch ein Bergmassiv mit beachtlichen Höhenunterschieden und damit deutliche Variationen in der Energiekonzentration.

Beide Landschaften verhalten sich außerdem komplementär zueinander. Was sich in der Orientierung als Schluchten ausprägt, hat seine Entsprechung in den kegelförmigen Bergkonstellationen in der Intention. Eine andere Denkweise wird in der Landvermessung praktiziert und gilt der Messung von Abständen zwischen unterschiedlichen Höhen. Wird zum Beispiel, eine unbekannte Landschaft untergestellt, müsste sich eine Orientierung auf die Höhenwerte einzelner Messungspunkte stützen können.

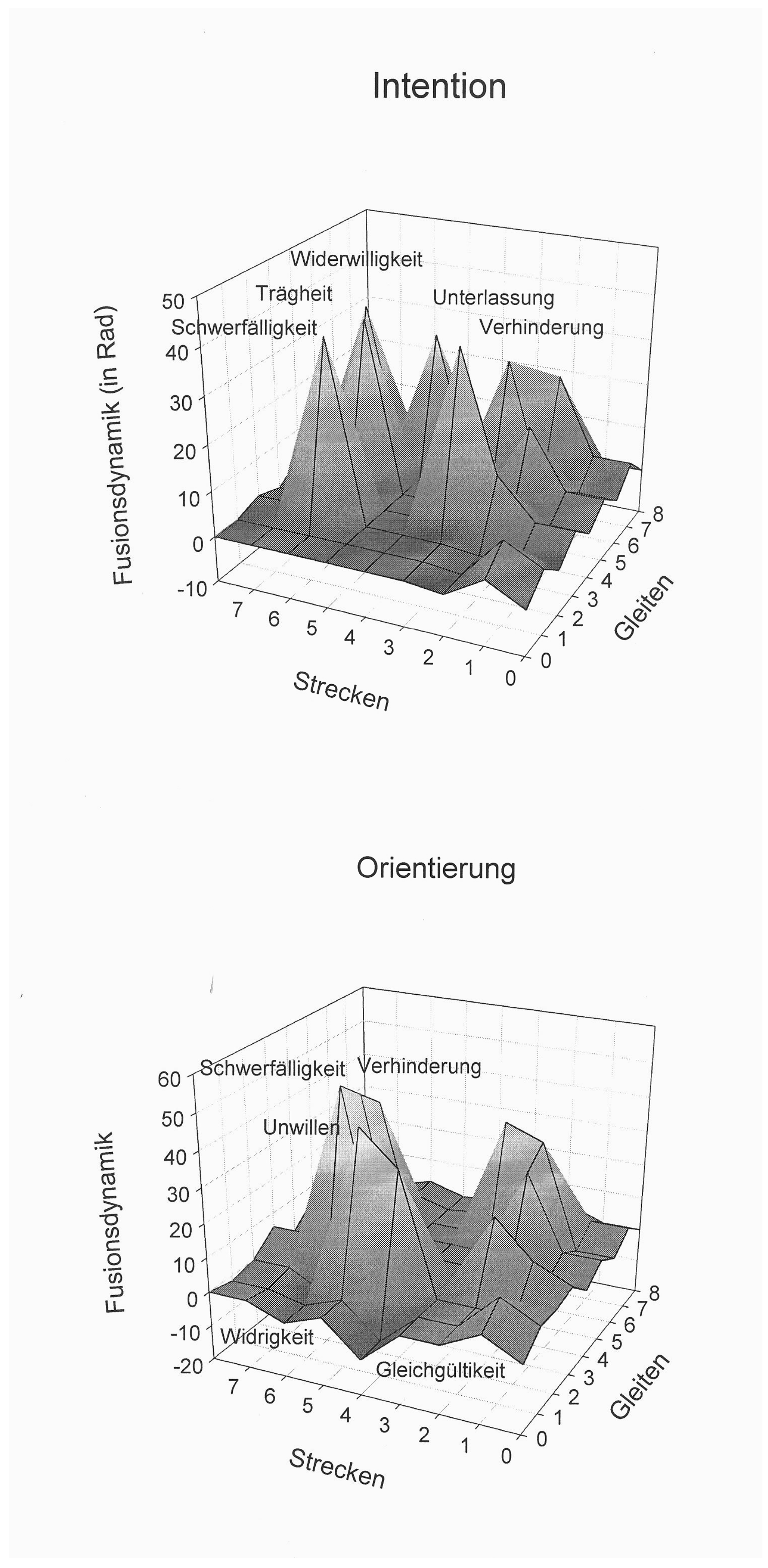


Abbildung 4 Gefaltete *Texträume*

Mit dem Ausgang in der Landvermessung ist jeder topologische Punkt (=Singularität) einer Landschaft durch die Knoten in der dazugehörigen Grundmatrix eindeutig definiert. In der Orientierung ist die Singularität (T_{O17}) mit dem Wert (~+50.19) verknüpft, während sich die Singularität (T_{O22}) durch den Wert (~-9.61) charakterisieren lässt. Werden die Werte für Höhen und Tiefen für eine eindeutige Bezeichnung verwendet, lässt sich der jeweils höchste oder niedrigste Punkt einer Landschaft beschreiben. Damit definiert der Punkt (T_{O22}) den niedrigsten Platz in dieser Landschaft. Auf der anderen Seite zeigt die Singularität (T_{O23}) mit dem Wert (~+34.85) jene Stelle an, an der das Landschaftsbild zwischen seine globale Singularität anzeigt. Diese Verhältnisse sind in den 3D-Graphen die gegeben topologischen Referenzpunkte.

Die topologische Vorstellung hat zum Beispiel mehrere Bergspitzen bekannt gemacht und die Spitzen mit Messungen markiert. Eigentlich könnte die gesamte geometrische Beschreibung mit dieser Darstellung abgeschlossen werden. Wie aus Abbildung 4 hervorgeht, ist der Beschreibungsprozess aber damit noch nicht zu Ende gekommen. Der Kontext der Textproduktion muss in den Benennungsprozess einbezogen werden können. Damit werden die bisher ausgeschlossenen semantischen Informationen aktualisiert. Es ist nämlich unausweichlich so, dass Textstrukturen in der Oberfläche, d. h. in der Textur eines Texts verankert sind. Deswegen sollen jetzt die graphembasierten semantischen Informationen der Tabelle 7 wieder zur Anwendung kommen.

Exkurs

Übung 14: Benennung der Singularitäten

Um die Charakteristik des produzierten Texts mit der Benennung der Faltungen fortsetzen zu können, muss die Verknüpfung von Graphemstrengen mit ihren Kantenwerten festgestellt werden. Mit Tabelle 17 soll dieser Prozesse verdeutlicht werden:

Die Transformation beginnt mit Größe (β_1), die mit dem Graphemstreng āSie sich die Einstellungó zusammengefügt ist. Aus diesem Grund muss der erste Graphemstreng durch die zweite Größe (β_2), die mit dem zweiten Streng āvon heutzutageó verknüpft ist, zu etwas Begrifflichen transformiert werden. Verallgemeinerungen besitzen also nicht mehr die gleiche physische Existenz wie ein Streng, der in die Verknüpfung eingegangen ist. Für die resultierende Fusion bietet sich *Einstellung* als Essenz der Fusion an.

In gleicher Weise vereinigen sich in Tabelle 17 die Graphemsequenzen āja nicht nuró von (β_4) mit (β_5), d. h. āunter den Gemeindeangestelltenó. Das Resultat dieser Transformation hat seine Beschreibung im Begriff *Trend* gefunden. Damit nun die Transformation einen Einfluss auf die nächst folgende Transformation nehmen kann, muss dieser Einfluss zu etwas Neuem führen.

In der Rückwärtsbindung, wird veranschaulicht auf welche Weise Einstellung durch Trend zu etwas transformiert, was den Grundton widerspiegelt. Hier bietet sich der Terminus ***Verbreitung*** als geeignetes Resultat der Transformation an.

Um zwischen alternativen Bezeichnungen wählen zu können, muss der interaktive Prozess den Grundton in dieser Bindung einfangen können. Da das Ergebnis keine direkte Entsprechung im physischen Kontext hat, muss die Transformation, die durch die Rückwärtsbindung verursacht wird, allemal zu einem virtuellen Ergebnis führen. Wie auch immer die Lösung ausfällt, die Arbeit mit den Graphemstrengen etabliert Transformationen, die zu Namen für die Knoten führen, die geeignet sind, Neuerscheinungen (eng. *Novelties*) zu charakterisieren.

Tabelle 17
Transformation von β-Variablen

Knoten	*Wert*	*Transformation*	*Knoten*	*Wert*	*Transformation*
1	5.5892	Sie sich die Einstellung	**T_{12}**	**4.10631**	**Vermeidung**
2	5.1471	von heutzutage	*T_9*	*34.6233*	*Selbst-Hinlänglichkeit*
T_1	**10.7363**	**Einstellung**	*T_{12}*	*4.10631*	*Vermeidung*
4	4.8042	ja nicht nur	**T_{13}**	**38.72961**	**Unterlassung**
5	6.1146	unter den Gemeinde angestellten	D	0	
T_2	**10.9188**	**Trend**	14	5.6834	mir doch völlig egal
T_1	*10.7363*	*Einstellung*	**T_{14}**	**5.6834**	**Gleichgültigkeit**
T_2	*10.9188*	*Trend*	*T_{13}*	*38.72961*	*Unterlassung*
T_3	**21.6551**	**Verbreitung**	*T_{14}*	*5.6834*	*Gleichgültigkeit*
D	0		**T_{15}**	**44.41301**	**Trägheit**
6	4.082	eben	D	0	
T_4	**4.082**	**Ebenheit**	15	5.7776	das gleiche Denken hier
T_3	*21.6551*	*Verbreitung*	**T_{16}**	**5.7776**	***Gleichheit***
T_4	*4.082*	*Ebenheit*	*T_{15}*	*44.41301*	*Trägheit*
T_5	**25.7371**	**Stimmung**	*T_{16}*	*5.7776*	*Gleichheit*
D	0		**T_{17}**	**50.19061**	**Schwerfälligkeit**
7	4.8984	ja mein Gehalt	D	0	
T_6	**4.8984**	**Einkommen**	12	-1.64499	(wie die Gemeinde Geld+ das+mir+ doch völlig egal)
T_5	*25.7371*	*Stimmung*	**T_{18}**	**-1.64499**	**Missklang**
T_6	*4.8984*	*Einkommen*	D	0	
T_7	**30.6355**	**Sicherheit**	3	-4.08791	An+es+ja nicht nur unter den Gemeindeangestellten
D	0		**T_{19}**	**-4.08791**	**Charakterisierung**
8	3.98978	ich	*T_{18}*	*-1.64499*	*Missklang*
T_8	**3.98978**	**Ichbezogenheit**	*T_{19}*	*-4.08791*	*Charakterisierung*
T_7	*30.6355*	*Sicherheit*	**T_{20}**	**-5.7329**	**Widrigkeit**
T_8	*3.98978*	*Ichbezogenheit*	T_{17}	50.19061	Schwerfälligkeit
T_9	**34.6233**	**Selbst-Hinlänglichkeit**	*T_{20}*	*-5.7329*	*Widrigkeit*
D	0		**T_{21}**	**44.45771**	**Verhinderung**
9	1.1304	Wege	D	0	
T_{10}	**1.1304**	**Möglichkeiten**	10	-9.60956	(wie die Gemeinde Geld+das +mir+doch völlig egal)
D	0		**T_{22}**	**-9.60956**	**Gleichgültigkeit**
13	2.97591	das+mir +doch völlig egal	*T_{21}*	*44.45771*	*Verhinderung*
T_{11}	**2.9759**1	**Widerwilligkeit**	*T_{22}*	*-9.60956*	*Gleichgültigkeit*
T_{10}	*1.1304*	*Möglichkeiten*	**T_{23}**	**34.848115**	**Unwillen**
T_{11}	*2.97591*	*Widerwilligkeit*			

Eingebaut in dem zugrunde liegenden Prozess ist ein Korrektionsmechanismus, der in den dynamischen Netzen verankert ist. Sollten nämlich diese interaktiven Transformationen einen Namen generieren, der zu weit von einem geeigneten Terminus abgekommen ist, wird der Prozess im weiteren Verlauf kaum eine sinnvolle Benennung zustande bringen können. Schließlich lässt sich auch noch vermuten, dass sich die Konfigurationen für bestimmte dynamische Netze nicht sinnvoll schließen können, da die ausschlaggebende Ringstruktur nicht entschlüsselt werden konnte.

Übung 15: Extraktion von Deskriptoren

Vorausgesetzt, dass die Transformationen erfolgreich abgeschlossen werden konnten, kann der Prozess noch einen Schritt weitergeführt werden. In Übereinstimmung mit Tabelle 18 etabliert dieser Schritt die Extraktion.

Tabelle 18

Extrahieren der Termini aus dem O-Netz

A-Komponente	*O-Komponente*		*Fusion*
Pendular	*Destination*	*Extrahiere*	*Wert*
T_1: 4 → 5	T_{O2}	Trend	8.5408
T_2: D → 6	T_{O4}	Ebenheit	4.4274
T_3: T_{A2} → T_{A1}	T_{O3}	Verbreitung	12.9682
T_4: D → 7	T_{O6}	Einkommen	3.8917
T_5: T_{A4} → T_{A3}	T_{O5}	Stimmung	16.8599
T_6: D → 8	T_{O8}	Ich-Bezogenheit	5.046333
T_7: T_{A6} →T_{A5}	T_{O7}	Sicherheit	21.906233
T_8: D → 11	T_{O22}	Gleichgültigkeit	4.9612
T_9: T_{A8} → T_{A7}	T_{O9}	Selbst-Hinlänglichkeit	26.867433
T_{10}: D → 12	T_{O11}	Missklang	3.272625
T_{11}: T_{A10} → T_{A9}	T_{O13}	Unterlassung	30.140058
T_{12}: D → 13	T_{O11}	Widerwilligkeit	*3.8936*
T_{13}: T_{A12} → T_{A11}	T_{O11}	Widerwilligkeit	34.033658
T_{14} D → 14	T_{O14}	Gleichgültigkeit	3.5168
T_{15}: T_{A14} → T_{A13}	T_{O15}	Trägheit	37.550458
T_{16}: 1 → 2	T_{O1}	Einstellung	1.753844
T_{17}: T_{A16} → T_{A15}	T_{O17}	Schwerfälligkeit	39.304302
T_{18}: 1 → 2	T_{O1}	Einstellung	0.0000
T_{19}: D → 3	T_{O19}	Charakterisierung	0.0000
T_{20}: T_{A19} →T_{A18}	T_{O20}	Widrigkeit	0.0000
T_{21}: T_D → 10	T_{O22}	Gleichgültigkeit	-0.86015
T_{22}: T_{A21} → T_{A20}	T_{O20}	Widrigkeit	-0.86015
T_{23}: T_{A22} → T_{A20}	T_{O21}	Verhinderung	38.444152

Die Termini, die aus dem (O) Netz kommen, werden nämlich für eine exakte Beschreibung der Knoten im (A) Netz gebraucht. Durch das Herausziehen von Termini lässt sich nämlich auch noch die Intention benennen. Ausgangspunkt für den Verknüpfungsprozess ist die Größe (α_4). Von hier aus geht der Transformationsweg (→) über die Größe (α_5). Die gekrümmten Linien, die den Weg im (A) Netz beschreiben, enden in der Singularität (T_{A1}). Sicherheitshalber wurden im Prozess die Indexbezeichnungen (T_A) und (T_O) eingeführt, um zwischen den Singularitäten von (A) und (O) unterscheiden zu können. Um ausfindig machen zu können, welcher Terminus den aufgefundenen Knoten in (A) am besten beschreibt, geht der Prozess über zu (O) und sucht die vergleichbaren Größen auf, was bedeutet, dass sie identisch mit den angegebenen Bezeichnungen sein müssen, d. h. ($\alpha_4 = \beta_4$).

Im Transformationsprozess beginnt die notwendige Pendelbewegung mit der Größe (β_4), die den veranlassten Pendelschwung in der Größe (β_5) enden lässt. Die Singularität, die den unmittelbar vorhergehenden Platz einnimmt, ist (T_{O2}). Diese wird nun herausgezogen und zur Beschreibung von (T_{A1}) mit der entsprechenden Singularität verbunden, nämlich *Trend*. Die nächste Größe (α_6) ist mit einem Platzhalter (D) vereinigt. Wenn nun das Pendel in (O) auf die Größe (β_6) schwingt, zeigt sich, dass die Singularität (T_{O4} = Ebenheit) unmittelbar

davor liegt. Folglich beschreibt dieser Terminus in (A) den Knoten (T_{A2}). Auf diese Weise setzen sich die Pendelbewegungen fort, was bedeutet, dass die Bewegung von (T_{A2}) auf (T_{A1}) hin folgt. Als Konsequenz wird der Terminus für (T_{O3}) zur Beschreibung von (T_{A3}) extrahiert, nämlich *Verbreitung*. Das Resultat aller Pendelbewegungen bringt mit sich, dass die gesamte terminologische Charakteristik für (A) integriert werden kann.

Kommentar: Nachdem nun das gesamte Verfahren abgeschlossen ist, kann es nochmals am Platz sein, sowohl auf das axiomatische Verhältnis zwischen Agent (A) und Objektiv (O) als auch auf die Tatsache hinzuweisen, dass nur dieses komplementäre Verhältnis zu einer sinnvollen Beschreibung von Agent und Objektiv führen kann.

Literatur

Bierschenk, B. (1978). Innehållsanalys som forskningsmetod (Content analysis as research method). *Kompendieserien*, No. 25. Malmö: School of Education.

Bierschenk, B. (1984). Steering mechanisms for knowability. *Cognitive Science Research*, 1. Lund University.

Bierschenk, B. (1991). The schema axiom as foundation of a theory for measurement and representation of consciousness. *Cognitive Science Research*, 38. Lund University.

Bierschenk, B. (1993). The fundamentals of perspective text analys*is. Cognitive Science Research*, 45. Lund University.

Bierschenk, B. (2001). Geometric foundation and quantification of the flow in a verbal expression. *Cognitive Science Research*, 81. Copenhagen University & Lund University.

Bierschenk, B. (2002). Real time imaging of the rotation mechanism producing interview-based language spaces. *Cognitive Science Research*, 83. Copenhagen University & Lund University.

Bierschenk, B. (2005). Differentiated limits for knowability. *Cognitive Science Research*, 97. Copenhagen University & Lund University.

Bierschenk, B. (2011). Functional text geometry: The essentials of Perspective Text Analysis. *Cognitive Science Research*, 101. Copenhagen University & Lund University.

Bierschenk, B. & Bierschenk, I. (1976). *Computer-based content analysis*. Lund: Gleerup.

Bierschenk, B., & Bierschenk, I. (1986a). Concept formulation. Part II. Measurement of formulation processes. *Cognitive Science Research*, 11. Lund University.

Bierschenk, B., & Bierschenk, I (1986b). Concept formulation: Part III. Analysis of mentality. *Cognitive Science Research*, 12. Lund University. (ERIC, ED 275 161)

Bierschenk, B., & Bierschenk, I. (1993) Perspektivische Textanalyse In: E. Roth (Hrsg.), *Sozialwissenschaftliche Methoden* (Ss.175-203). München: Oldenbourg Verlag.

Bierschenk, B., & Bierschenk, I. (2011). Functional text geometry. The essentials of Perspective Text Analysis. *Cognitive Science Research*, 101. Copenhagen University & Lund University.

Bierschenk, B., Bierschenk, I., & Helmersson, H. (1996). Die Ökologie des Sprachraums In: W. Bos, & C. Tarnai (Hrsg.), *Computerunterstützte Inhaltsanalyse in den Empirischen Sozialwissenschaften. Theorie - Anwendung - Software* (Ss. 11-31). München: Waxmann.

Bierschenk, I. (1977). Datorbaserad innehållsanalys: Kodningsmanual [Computer-basierte Inhaltsanalyse: Codierungsmanual]. *Pedagogisk dokumentation, 52*. Lund University, Malmö: Department of Educational and Psychological Research. (In Swedish)

Bierschenk, I. (1984). The schematism of natural language. *Cognitive Science Research*, 2. Lund University.

Bierschenk, I. (1992). An excursion into the ecological co-ordinates of language space. *Cognitive Science Research*, 43. Lund University.

Bierschenk, I. (1999). The essence of text. A dialogue on Perspective Text Analysis. *Cognitive Science Research*, 70. Copenhagen University & Lund University.

Bierschenk, I. (2011a). Ett ekologiskt perspektiv på språk och textanalys. [Eine ökologische Perspektive auf die Sprache und die Textanalyse]. *Cognitive Science Research*, 98. Copenhagen University & Lund University. (In Swedish)

Bierschenk, I. (2011b). Applications of perspective text analysis. A thematic overview. *Cognitive Science Research*, 99. Copenhagen University & Lund University.

Bierschenk, I., & Bierschenk, B. (2004). Diagnose der Leistungsheterogenität durch die Perspektivische Textanalyse: VERTEX. In: W. Bos, Lankes, E.-M., Plaßmeier, N., & Schwippert, K. (Hrsg.), *Heterogenität: Eine Herausforderung an die empirische Bildungsforschung* (Ss. 16-28). Münster: Waxmann.

Bierschenk, I., & Bierschenk, B. (2011). Perspective Text Analysis: Tutorial to Vertex. *Cognitive Science Research*, 100. Copenhagen University & Lund University.

Bos, W., & Tarnai, C. (1989). *Angewandte Inhaltsanalyse in empirischer Pädagogik und Psychologie*. Münster: Waxmann Wissenschaft.

Bos, W., & Tarnai, C. (1996). *Computerunterstützte Inhaltsanalyse in den empirischen Sozialwissenschaften. Theorie, Anwendung, Software*. Münster: Waxmann Wissenschaft.

Greene, B. (1999). *The elegant universe. Superstrings, hidden dimensions, and the quest for the ultimate theory*. New York: W. W. Norton & Company.

Hestenes, D. (1986/1993). *New foundations for classical mechanics*. Dordrecht: Kluwer Academic.

Mackenzie, D. (1997). Through the looking glass. In arithmetic 5 and 7 can be added in any order to yield 12. When order does matter, you have entered the strange, disorientating world of noncommutativity. *The Sciences*, *37*(3), 32-37.

Pinker, S. (2002). *The blank slate. The modern denial of human nature*. London: Penguin Books.

SigmaPlot (2008). *Exact graphs for exact science. Userȏs manual* (Version 11). Chicago: SPSS Inc.

Wales, D. J. (2003). *Energy landscapes. With applications to clusters, biomolecules and glasses.* Cambridge: Cambridge University Press.

Accepted March 07, 2014